UiT The Arctic University of Norway

Faculty of Science and Technology

Department of Physics and Technology

Advancing Deep Learning for Marine Environment Monitoring

Changkyu Choi

A dissertation for the degree of Philosophiae Doctor - June 2023

To Yoon.

Abstract

Marine environment monitoring has become increasingly significant due to the excessive exploitation of oceans, which detrimentally impacts ecosystems. Deep learning provides an effective monitoring approach by automating the analysis of vast amounts of observed image data, enabling stakeholders to make informed decisions regarding fishing quotas or conservation efforts.

The success of deep learning is often attributed to its capacity to extract relevant features from data, without the need for handcrafted rules or heuristics. However, this capability is not without limitations, as the intricate feature extraction process of deep learning-based systems poses fundamental challenges.

A lack of annotated data presents an inherent challenge for deep learning. The widespread success of deep learning has primarily relied on the ample availability of annotated data, while deep learning models encounter difficulties when learning from limited annotations. However, obtaining annotated data is expensive, particularly in the context of marine environment monitoring, as it is often a manual process demanding the expertise of domain specialists.

Another challenge of deep learning is a lack of explainability. The black-box nature of deep learning models can make it difficult to understand how they arrive at their decisions. This hinders their adoption in critical decision-making processes, as stakeholders may be hesitant to trust models whose decision-making rationale is not transparent or interpretable.

To address the challenges and further advance deep learning methodologies, this thesis proposes three novel deep learning methods, highlighting marine environment monitoring as an application domain. The dependence on annotated data is addressed through two novel semi-supervised methods demonstrated in different image tasks. The central operational logic in both methods entails alternating between supervised learning and unsupervised deep clustering within a single network, merging data structure uncovered through unsupervised clustering with a small amount of ground-truth class information. Both methods employ multi-frequency echosounder data to demonstrate their effectiveness in marine environment monitoring, outperforming conventional approaches.

Moreover, a new explainable deep learning method is proposed to address the lack of explainability. This method generates explanations for its decisions while adhering to user-centered explanation principles, such as minimality, sufficiency, and interactivity. The information-bottleneck framework provides a theoretical ground to pursue minimality and sufficiency, while interactivity is accomplished by integrating additional domain knowledge into the training process, enabling the generated explanations to evolve accordingly. The method is validated using a variety of marine image datasets, encompassing multi-frequency echosounder data and seal pup images on sea ice.

While the monitoring of marine environments is a significant focus, the primary aim of the thesis is to contribute to the advancements of deep learning methodologies. As such, the proposed methods are designed to be generic and possess the potential for broader applicability across various domains. We believe that the methods presented in this thesis hold the promise of fostering a more effective, user-centered, and transparent approach to deep learning, as well as facilitating our efforts to preserve the marine environment and promote sustainable ocean stewardship.

Acknowledgements

A number of people deserve recognition and appreciation for their support and guidance throughout the completion of this doctoral thesis.

First and foremost, I would like to express my deepest appreciation to my main supervisor, Professor Robert Jenssen, for his unwavering support, patience, and invaluable expertise throughout my doctoral journey. I must confess that one of my primary motivations for embarking on the long journey from Seoul to Tromsø was to learn from his expertise. I am grateful for the opportunity to work with him as the main supervisor for both my master's and doctoral thesis, and I wouldn't have been able to complete this thesis without his continuous encouragement and guidance. His approach as a researcher and maturity as an individual continue to inspire me, and I aspire to emulate them.

I would like to extend my heartfelt appreciation to my co-supervisors, Dr. Arnt-Børre Salberg and Associate Professor Michael Kampffmeyer, for their continuous mentorship at the intersection of two domains of deep learning and marine environmental monitoring. Working with such an excellent supervision team has been a great fortune for me, since they have provided me with the knowledge and insights necessary to navigate the complexities of this journey.

I would also like to thank the members of my thesis committee, Professor Morten Goodwin, Associate Professor Vedrana Dahl, and Associate Professor Benjamin Ricaud, for their time and effort in reviewing and evaluating my thesis.

My sincere thanks go to my co-authors, Assistant Professor Shujian Yu, Dr. Nils Olav Handegard, Line Eikvil, and Olav Brautaset, who have collaborated with me on this project. Their intellectual contributions, critical analysis, and dedication have played an essential role in the advancement of this research.

I would also like to acknowledge my colleagues and fellow researchers at UiT machine learning group, who have provided a stimulating and collaborative environment for my research. Their camaraderie, suggestions, and discussions have been instrumental in shaping my ideas and refining my work. Almost as impressive as their research excellence is their great sense of humor, which has made working with this group a truly enjoyable experience. Furthermore, I am grateful for their kind invitation to a gathering, where I was warmly welcomed. It was a wonderful experience to feel the warmth of northern Norway and to be a part of this welcoming community.

Last but not least, I am greatly indebted for the constant love, understanding, and encouragement provided by my family and friends throughout my academic journey, which has been a crucial source of strength and motivation. As a symbol of my appreciation, I dedicate this thesis to them.

Abbreviations

CNN	Convolutional neural networks
ATC	Acoustic target classification
MLP	Multi-layer perceptron
FCNN	Fully connected neural networks
BN	Batch normalization
ReLU	Rectified linear unit
XOR	Exclusive or
SIFT	Scale-invariant feature transform
GLOH	Gradient location orientation histogram
SVM	Support vector machine
FCN	Fully convolutional networks
XAI	Explainable artificial intelligence
IB	Information bottleneck
CE	Cross entropy
Grad-CAM	Gradient-weighted class activation mapping
LRP	Layer-wise relevance propagation
DeepLIFT	Deep learning important features
LIME	Local interpretable model-agnostic explanations
RISE	Randomized input sampling for explanation
SENN	Self-explainable neural networks
FCN	Fully convolutional networks
VGG	Visual geometry group
PASCAL	Pattern analysis, statistical modelling, and computational learning
VOC	Visual object classes

List of Publications

Included papers

This thesis is based on the following original journal papers:

I. Changkyu Choi, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, Olav Brautaset, Line Eikvil, and Robert Jenssen.**"Semi-supervised Target Classification in Multi-frequency Echosounder Data"**, in *ICES Journal of Marine Science*, vol. 78, no. 7, Oct. 2021.

II. Changkyu Choi, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, and Robert Jenssen.**"Deep Semi-supervised Semantic Segmentation in Multi-frequency Echosounder Data"**, in *IEEE Journal of Oceanic Engineering*, vol. 48, no. 2, 2023.

III. Changkyu Choi, Shujian Yu, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, and Robert Jenssen. **"Deep Deterministic Information-Bottleneck Explainability on Marine Image Data"**, submitted to *Pattern Recognition*, 2023.

Other papers

The following original papers and poster presentation also contribute to the thesis, but are not included:

4. Handegard, Nils Olav, Lars Nonboe Andersen, Olav Brautaset, Changkyu Choi, Inge Kristian Eliassen, Yngve Heggelund, Arne Johan Hestnes, Ketil Malde, Håkon Osland, Alba Ordonez, Ruben Patel, Geir Pedersen, Ibrahim Umar, Tom Van Engeland, and Sindre Vatnehol. **"Fisheries Acoustics and Acoustic Target Classification-Report from the COGMAR/ CRIMAC Workshop on Machine Learning Methods in Fisheries Acoustics."** *Rapport fra Havforskningen*, 2021.

5. Changkyu Choi, Filippo Maria Bianchi, Michael Kampffmeyer, and Robert Jenssen. **"Short-Term Load Forecasting with Missing Data using Dilated Recurrent Attention Networks"**, *In Proceedings of the Northern Lights Deep Learning Conference (NLDL)*, vol. 1, 2020.

6. Changkyu Choi, Michael Kampffmeyer, and Robert Jenssen. **"A Robustness Analysis of Personalized Propagation of Neural Prediction"**, *poster presentation at Northern Lights Deep Learning Conference (NLDL)*, Jan. 2020.

Contents

List of Figures

1 Introduction

Marine environment monitoring involves observing the physical, chemical, and biological characteristics of the marine environment and deriving knowledge from these observations [7]. The importance of this monitoring has grown significantly due to excessive exploitation of oceans and their resources, leading to adverse effects on ocean ecosystems [8, 9]. Consequently, there is a growing awareness of the need to protect and conserve the marine environment for future generations, with increasing efforts to adopt sustainable practices [10].

Deep learning has been introduced as a means to monitor the marine environment more effectively, aiming to extract reliable knowledge through the automated analysis of vast amounts of observed data [11]. Deep learning-based methods have demonstrated their ability to automate the analysis process for marine images, such as marine-coastal images from unmanned aerial vehicles [12], images from underwater trawls [13], remote sensing images [14], and underwater acoustic images [7]. This automation has made the process more efficient and less time-consuming compared to conventional heuristic approaches [15].

Recent research suggests that utilizing deep learning-based methods can provide a more precise and comprehensive analysis of the marine environment and its ecosystems compared to traditional manual methods. Examples include inspecting water quality [16, 17], estimating fish populations [6, 15, 18], and monitoring coral reef health [19, 20]. These enhanced monitoring capabilities enable stakeholders to identify emerging threats to oceans and their ecosystems, leading to more informed management decisions related to fishing quotas, conservation efforts, and coastal development [11].

The aim of this thesis is to advance deep learning methodologies for marine environment monitoring by addressing some of the challenges in automated analysis and interpretation of marine image data. These challenges will be briefly outlined in the following section and addressed in greater detail in the included papers.

1.1 Key challenges

There has been made significant progress on automation of marine image data analysis thanks to thanks to advances in deep learning and computer vision. However, challenges related to both the deep learning methods and the specific nature of marine image data remain. In the context of these diverse challenges, the key challenges addressed in this thesis are as follows:

Complex data representation Marine image data of major interest in this thesis is multi-frequency echosounder data [21]. This data is collected using an echosounder that emits acoustic pulses and captures echoes that backscatter from underwater objects, such as marine organisms or the seabed, in a non-invasive manner. Echosounders, being highly sensitive instruments, are capable of detecting even the smallest amounts of acoustic backscatter. This sensitivity, however, makes them vulnerable to external sources of unwanted noise, such as acoustic and electrical noise [22]. These noise sources can include electrical or mechanical interference, acoustic cross-talk from high-energy pulses from other acoustic systems, and excessive acoustic attenuation that reduces backscatter. The impact of these noise sources on data quality can vary depending on the measurement's climate, complicating the analysis process [23].

Another challenge in analyzing echosounder data is the significant class imbalance at the pixel level, which poses difficulties for statistical analysis, including deep learning-based methods. This imbalance typically occurs because some classes, such as fish species, have very few samples compared to other classes, such as background classes containing water. This class imbalance can cause models to be biased toward the majority class, leading to poor performance when identifying or predicting minority classes.

Lack of annotated data The widespread success of deep learning methods has largely depended on the increasing availability of annotated datasets [24]. However, in the context of marine environment monitoring, obtaining annotated data can be a significant challenge. The collection of marine images often lacks annotated information, and annotating such data requires the expertise of multiple domain specialists, which can be both time-consuming and costly [15, 25]. Furthermore, in cases where the data is noisy or challenging, domain specialists may disagree on the correct annotation, making the annotation process even more difficult. Consequently, the lack of annotated data poses a major challenge for marine environment monitoring, as deep learning models struggle to learn from limited or inconsistent annotations.

Lack of explainability Another challenge faced by deep learning is the lack of explainability [26]. While deep neural networks have demonstrated impressive performance on a wide range of marine environment monitoring tasks [6, 15–18], their black-box nature can make it difficult to understand how they arrive at their decisions [27]. This challenge is not exclusive to the domain of marine sciences and is also a significant obstacle in other fields that place a high value on trust and accountability [28–32]. The lack of explainability in deep learning models hinders their adoption in critical decision-making processes, as stakeholders may be hesitant to rely on models whose decision-making rationale is not transparent or interpretable.

1.2 Key objectives

Addressing the challenges outlined in the previous section is crucial to advancing both deep learning and marine environment monitoring. Therefore, this thesis focuses on developing generic deep learning methods that can be applied to various domains, with marine environment monitoring serving as a particularly significant application domain. With this in mind, the key objectives of this thesis are formulated as follows:

1. To propose a novel deep learning method that can effectively perform with limited annotated data.

2. To investigate the potential of the proposed method in addressing image-based tasks.

3. To develop a new explainable deep learning method that generates explanations tailored to the needs and preferences of the intended users.

4. To evaluate the effectiveness of the proposed methods within the context of marine environment monitoring.

By achieving these objectives, this thesis aims to contribute to the advancement of marine environment monitoring through deep learning-based methods. Ultimately, the goal is to collaboratively connect these two mature fields, fostering mutual advancements and driving positive change in both domains.

1.3 Key solutions

This thesis consists of three deep learning papers that address the objectives previously outlined. Papers I and II tackle the challenge of limited annotated data, while Paper III is dedicated to enhancing the explainability of deep learning models.

In Paper I, a generic semi-supervised deep learning method for image classification is introduced, which combines the strengths of supervised learning and unsupervised deep clustering [33]. This method effectively integrates the data structure revealed through unsupervised clustering with the ground-truth class information present in a limited number of training samples. Building upon the groundwork established by Paper I, Paper II extends the semi-supervised approach to encompass the more complex image task of semantic segmentation. To demonstrate the effectiveness of these methods, both papers employ real-world multi-frequency echosounder data [34]. The proposed approaches not only outperform traditional manual methods but also efficiently handle the extreme class imbalance cases that are commonly encountered in echosounder data.

Paper III presents a novel explainable deep learning method that addresses the challenge of model explainability. The proposed method simultaneously generates explanations for its decisions while adhering to the principles of user-centered explanation, which include minimality, sufficiency, and interactivity. The minimality and sufficiency principles are pursued based on the information-bottleneck (IB) framework [35, 36], a well-formulated mathematical framework grounded in information theory [37, 38]. Interactivity is achieved by incorporating additional domain knowledge into the training process so that the generated explanation can evolve based on it. The proposed method is evaluated on multiple marine image datasets, including multi-frequency echosounder data [39] and image data of seal pups on sea ice [40].

1.4 Brief summary of included papers

This section presents a list of the papers included in this thesis, along with a summary of each paper. Additionally, lists of other published articles and academic presentations during this PhD project are included in Section 1.5 and Section 1.6, respectively. Figure 1.1 provides an overview of the topics covered as part of this thesis.

I. Changkyu Choi, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, Olav Brautaset, Line Eikvil, and Robert Jenssen.**"Semi-supervised Target Classification in Multi-frequency Echosounder Data"**, in *ICES Journal of Marine Science*, vol. 78, no. 7, Oct. 2021.

II. Changkyu Choi, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, and Robert Jenssen.**"Deep Semi-supervised Semantic**

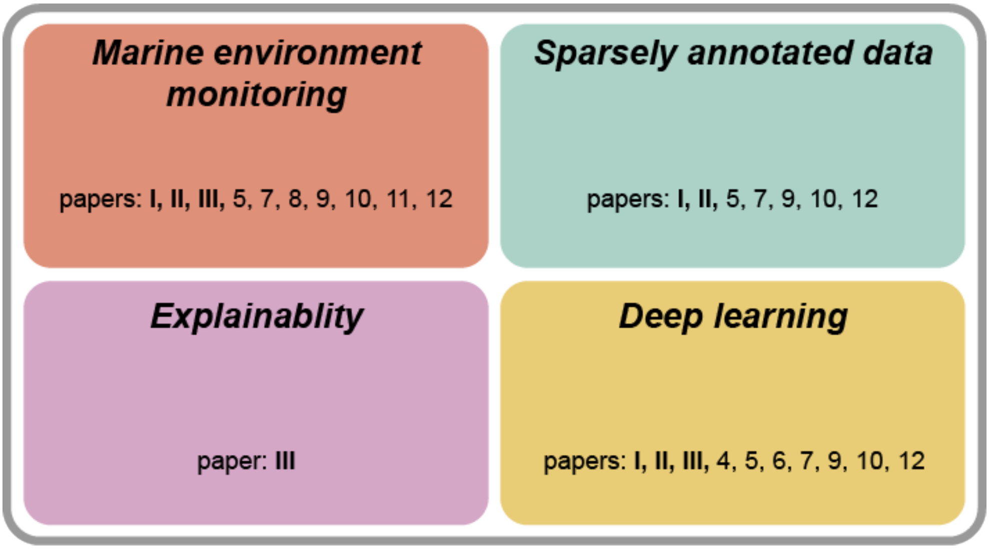

Marine environment monitoring
papers: I, II, III, 5, 7, 8, 9, 10, 11, 12
Sparsely annotated data
papers: I, II, 5, 7, 9, 10, 12
Explainablity
paper: III
Deep learning
papers: I, II, III, 4, 5, 6, 7, 9, 10, 12

Figure 1.1: An overview of the topics addressed in this thesis.

Segmentation in Multi-frequency Echosounder Data", in *IEEE Journal of Oceanic Engineering*, vol. 48, no. 2, 2023.

III. Changkyu Choi, Shujian Yu, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, and Robert Jenssen. **"Deep Deterministic Information-Bottleneck Explainability on Marine Image Data"**, submitted to *Pattern Recognition*, 2023.

Paper I presents a novel method for semi-supervised deep learning, which utilizes both annotated and unannotated data samples within a single convolutional neural network. The proposed method involves two objectives: a clustering objective and a classification objective, which are optimized in an alternating manner. The clustering objective aims to uncover the underlying structure of the entire training data in an unsupervised manner, while the classification objective enforces consistency between the underlying structure sought by the clustering objective and the available annotated data samples in a supervised manner. The proposed method is evaluated on image classification of multi-frequency echosounder data, and the results demonstrate its effectiveness.

Paper II proposes a novel method called PredKlus, which is a generalization of the semi-supervised deep learning method proposed in Paper I applied to the semantic segmentation task. In practice, the fisheries and aquatic industry has a particular interest in semantic segmentation, as it enables noninvasive estimation of marine organism abundance and observation of the underwater environment on a large scale. However, the high degree of class imbalance in semantic segmentation, where the background class accounts for approximately 99 percent of total pixels, poses a significant challenge [6]. To address this issue, the proposed method introduces a class-balancing technique based on the model's prediction, in addition to the alternating optimization proposed in Paper I. The proposed semi-supervised segmentation method is evaluated through experiments, achieving comparable results to the standard supervised semantic segmentation method while using only a small amount of annotated data.

Paper III proposes DIB-X, a novel self-explainable method that places an emphasis on user-centered explanations, which are represented by the following principles: minimality, sufficiency, and interactivity [41]. These principles are mathematically formulated in the objective function, enabling the network to learn the principles during optimization. The minimality and sufficiency principles are formulated through the information-bottleneck (IB) framework [35], which aims to find the optimal balance between their trade-off relationship. Distinctly, DIB-X directly quantifies the minimality principle using the recently proposed matrix-based R'enyi's α-order entropy functional [36], circumventing the need for variational approximation. The interactivity principle is realized by incorporating existing domain knowledge as prior explanations, fostering explanations that align with established domain understanding. Empirical results on two marine environment monitoring datasets demonstrate the effectiveness of the proposed method.

1.5 Other papers

In addition to the papers presented in this thesis, there are several works that, although not included, have made academic contributions to the field. These works offer alternative perspectives, methodologies, and insights that have enriched the understanding and advancement of deep learning methods within the realm of marine environment monitoring. Furthermore, the results were disseminated as part of an interview by Tekfisk, a magazine specializing in marine technology, which resulted in a featured article (title: Arbeidet hans sparer forskerne for store ressurser, og åpner for et selvstyrt fiskeri (2021), journalist: Ketil Svendsen).

4. Nils Olav Handegard, Lars Nonboe Andersen, Olav Brautaset, Changkyu Choi, Inge Kristian Eliassen, Yngve Heggelund, Arne Johan Hestnes, Ketil Malde, Håkon Osland, Alba Ordonez, Ruben Patel, Geir Pedersen, Ibrahim Umar, Tom Van Engeland, and Sindre Vatnehol. "Fisheries Acoustics and

Acoustic Target Classification", in *COGMAR/CRIMAC Workshop on Machine Learning Methods in Fisheries Acoustics"*, 2021.

5. Changkyu Choi, Filippo Maria Bianchi, Michael Kampffmeyer, and Robert Jenssen. "Short-Term Load Forecasting with Missing Data using Dilated Recurrent Attention Networks", in the proceedings of the *Northern Lights Deep Learning Conference (NLDL)*, 2020.

1.6 Presentations

6. Changkyu Choi, Michael Kampffmeyer, and Robert Jenssen. "A Robustness Analysis of Personalized Propagation of Neural Prediction", in a poster presentation at the *Northern Lights Deep Learning Conference (NLDL)*, 2020.

7. Changkyu Choi, "Semi-supervised Target Classification in Multi-frequency Echosounder data", in *COGMAR and CRIMAC Workshop on Fisheries Acoustics Classifiers*, 2020.

8. Changkyu Choi, "Deep Semi-supervised Target Classification in Multi-frequency Echosounder Data", in an oral presentation at *Norwegian Society for Image Processing and Machine Learning (NOBIM) Workshop*, 2021.

9. Changkyu Choi, "Semi-supervised Semantic Segmentation in Multi-frequency Echosounder Data", in a poster presentation at the *Northern Lights Deep Learning Conference (NLDL)*, 2021.

10. Nils Olav Handegard, Olav Brautaset, Changkyu Choi, Tomasz Furmanek, Arne Johan Hestnes, Espen Johnsen, Alba Ordonez, Ingrid Utseth, Sindre Vatnehol, and Geir Huse. "Developing and Deploying Machine Learning Methods for Acoustic Data", in *Workshop WGFAST - Working Group on Fisheries Acoustics Science and Technology*, 2022.

11. Changkyu Choi. "Segmenting Multi-frequency Marine Acoustic Data in a Semi-supervised Fashion", in a poster presentation at the *Northern Lights Deep Learning Conference (NLDL)*, 2022.

1.7 Reading guide

This thesis is organized into three main parts: *methodology and context*, *summary of research and concluding remarks*, and *included papers*. In the first part, three chapters provide the relevant background information for understanding the papers. Chapter 2 offers a brief introduction to marine environment monitoring, focusing on acoustic target classification. Chapter 3 covers the fundamental aspects of deep learning, with a spotlight on convolutional neural networks. Chapter 4 delves into advanced deep learning topics, such as semi-supervised deep learning and explainable deep learning. The second part consists of four chapters. Chapters 5 through 7 present a concise overview of the scientific contributions of each paper in the thesis. Chapter 8 contains the concluding remarks, and discusses the limitations of the research as well as potential future work. Finally, the third part of the thesis includes the three research papers that form the basis of this work.

1.8 Open science

Reproducibility plays a crucial role in advancing scientific progress [42]. In the field of deep learning, promoting open research can be achieved by sharing resources, such as code and data, or providing comprehensive details required to replicate experiments. To enhance the transparency of the research presented in this thesis, we have made the code and related resources publicly accessible. These materials can be found in the SFI Visual Intelligence Github repository (*github.com/SFI-Visual-Intelligence*) and are thoroughly described within the context of each respective research paper.

In addition to promoting open science through code and resources, we also strive to make our datasets publicly available. At present, the marine environmental observation data used in this thesis is not fully accessible to the public, as various organizations have vested interests in the release of this data. Nevertheless, we are committed to keeping the Github repository updated with information on how to access the data as the study progresses and as circumstances permit.

Part I

Methodology and Context

2 | Marine environment monitoring

This chapter offers insight into marine environment monitoring and the motivations behind the deep learning methods proposed in the presented papers, with a focus on acoustic target classification (ATC). Section 2.1 introduces an overview of ATC. Section 2.2 describes common analysis methods for ATC, including conventional approaches and their limitations, as well as more advanced methods based on deep learning. Lastly, Section 2.3 presents the marine environment monitoring data studied in the presented papers, which includes echosounder data from the sandeel survey [34] and aerial images of seal pups on sea ice [40].

2.1 Acoustic target classification

The marine environment is renowned for its rich biodiversity [43]. This can be explored using echosounder data that offers a non-invasive, large-scale visualization of the underwater environment [21]. Echosounder data consists of the echoes of acoustic pulses emitted by the echosounder, reflected from underwater objects such as marine life, converted into electrical signals, and digitally stored in the echosounder.

Echosounder data is displayed as a two-dimensional plot, with the vertical axis representing water depth (spatial resolution in centimeters) and the horizontal axis representing time (millisecond-scale resolution) [39]. Echosounder data, measured across multiple frequency channels, exhibits different backscatter patterns at varying frequencies, helping to distinguish between marine life species [44].

ATC is a rapidly advancing field of marine science, which aims to discern the size, shape, and composition of marine life from echosounder data and provides insights into their behavior, distribution, and abundance [2]. Moreover, ATC facilitates monitoring changes in populations over time [45, 46], making echosounder data a strategic asset for fisheries management and ecosystem conservation [44, 47, 48]. While ATC primarily focuses on the analysis methods applied to echosounder data, it also encompasses the entire process,

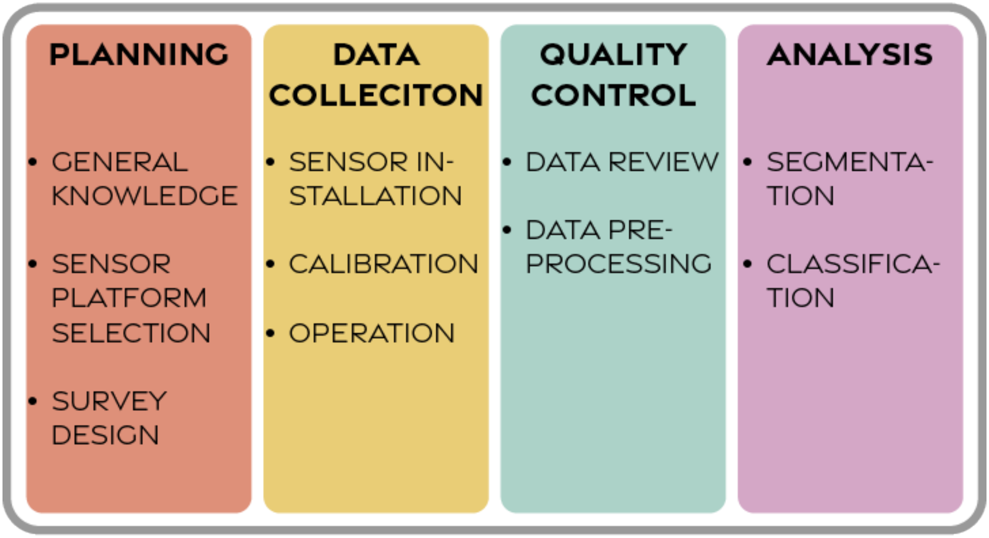


Figure 2.1: Workflow for acoustic target classification.

including survey planning and data collection, providing a comprehensive understanding of the underwater environment [21].

2.1.1 Overview

ATC consists of four main phases: planning, data collection, data quality control, and analysis [21]. Figure 2.1 illustrates the workflow of ATC, where the majority of the contribution of deep learning-based methods is seen in the analysis phase. As shown in the figure, the success of the analysis phase is highly dependent on the processes that precede it. Therefore, a holistic understanding of the ATC field is necessary to explore the potential directions that deep learning methods can take. By gaining a better understanding of the various steps in ATC, we can identify opportunities for applying deep learning to improve classification performance and enable the analysis of large and complex acoustic datasets.

Planning In the planning phase, the initial step involves defining the objective of ATC, as it influences subsequent decisions for survey design [49]. For example, if the objective is a stock assessment, identifying the target fish species and the target size becomes essential. Once the research objective is established, gathering relevant general knowledge is the next step. This may include understanding biological aspects [50–52] such as spawning behavior, migration patterns, historical information, temporal and spatial distribution, as well as scattering characteristics [53] and environmental conditions during data collection. Utilizing existing literature, local knowledge from fishers, and other resources can be beneficial for informing survey design [54]. Furthermore, it is essential to consider factors such as permit requirements, ethical approval, and the availability of personnel with appropriate expertise.

The choice of a sensor platform can greatly influence the quality of collected data, as different platforms are optimized for specific purposes and target species. Echosounders, originally used as ship-borne sensors, are mounted on the hull, drop keel, or pole of large platforms like research vessels [22, 55]. These platforms offer the advantage of conducting large-scale spatial surveys of fish and plankton distribution within short survey times [56]. However, due to the high cost of research vessels, researchers are investigating various alternative methods for collecting echosounder data. These alternatives include the use of echosounders on smaller platforms, such as fishing vessels [57], autonomous underwater gliders [58], and autonomous surface vehicles [59]. These innovative platforms provide opportunities to carry out acoustic surveys in areas that are challenging to access or monitor using traditional ship-borne echosounders.

Data collection The primary principle guiding the data collection phase is ensuring the physical and spatial comparability of echosounder data gathered at varying frequencies across different surveys [60]. This essential goal is closely tied to the installation, calibration, and operation of the echosounder system.

To achieve physical comparability, strict measures must be taken to maintain a high level of consistency in the physical parameters of the equipment [39], including settings, frequency channels, environmental conditions, and noise reduction techniques during data collection. Maintaining this consistency ensures that data can be directly compared and analyzed without significant deviations caused by changes in equipment or measurement processes.

On the other hand, spatial comparability demands that data collected from the same location be geographically comparable, regardless of the instruments or settings used during data collection [21]. This requires that measurements have equivalent spatial resolution, pulse lengths, pulse shapes, and comparable sampling volumes.

Quality control During the data quality control phase, the primary goal is to remove spurious patterns and minimize the variability of collected echosounder data for subsequent analysis [39]. Echosounders can potentially record unwanted signals, noise, and other inconsistencies that necessitate proper identification and handling. Thus, a comprehensive understanding of various types of noise and unwanted signals is essential for choosing the most appropriate denoising methods during data preprocessing [22].

In echosounder data, noise refers to uncorrelated interference [61], such as internal noise, platform-related noise, or asynchronous electronic or acoustic interference. Biological sources, like clicks and vocalizations from marine life [62], can also generate noise. Unwanted signals are backscatters from non-targeted objects that correlate with the transmit pulse, such as air bubbles, the seabed, or non-targeted biological organisms [63].

Analysis The analysis phase is the main focus of this thesis, during which we aim to transform the collected echosounder data into information and gain insights from it. This will be further discussed in the next section.

2.2 Analysis methods

During the analysis phase, the observed backscattered patterns are analyzed to determine which targets are present in the surveyed area. Either comparing the patterns with the modeled scattering characteristics or extracting features within the observed patterns, it is possible to classify the targets based on their acoustic properties [2, 64]. This approach enables researchers to identify and differentiate between different types of underwater targets, which can be used for a range of applications such as fisheries management [65] and habitat mapping [66].

2.2.1 Conventional approach

Conventional approaches are based on heuristics, in which an expert analyst manually identifies targets [22]. Depending on how the expert formulates the classification criteria, there are two primary directions for analyzing echosounder data, namely scattering model-based approaches [67, 68] or empirical approaches [60, 69].

Scattering model-based approach In the scattering model-based approach, echosounder data is analyzed through a pre-built scattering model for the target, describing the interaction between the transmitted acoustic wave and objects such as a target fish species within the water [68]. Figure 2.2 shows an example of the scattering pattern for a swimbladdered fish. The analysis is enabled through the comparison of measured backscattered patterns with the scattering model. This approach has been used in the early fisheries acoustics [22] and remains relevant when the target species and its scattering properties are well studied [67].

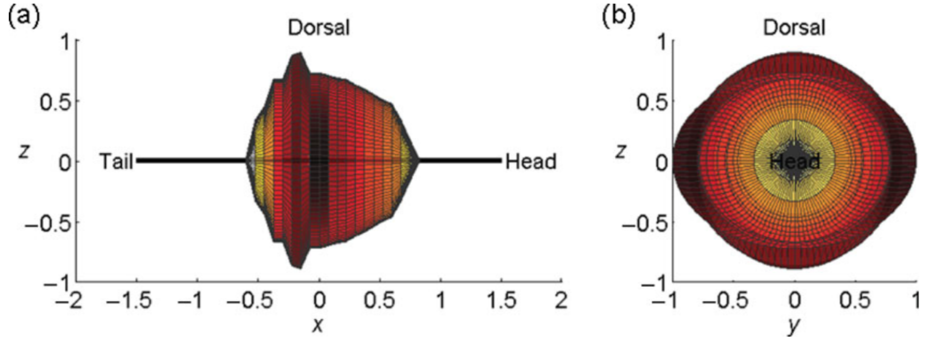


Figure 2.2: (a) Lateral and (b) head-on perspectives of a generic swimbladdered fish scattering-directivity pattern model. Example adapted from [1].

The scattering model-based approach offers advantages in acoustic fisheries [21]. It allows for the simulation of classification processes using controlled input parameters and variables, thereby facilitating the assessment of efficiency, effectiveness, robustness, and uniqueness [21]. Moreover, it enables validation and theoretical interpretation of the empirical approach, which will be described next [67].

Empirical approach The empirical approach does not rely on the scattering models of the target. Instead, this approach searches for features within the observed echosounder data that can be used to distinguish the target from others. Given the complex nature of the underwater environment, empirical approaches can be efficient in scenarios where the collected backscattering patterns of the target are relatively diverse, offering greater flexibility compared to scattering model-based approaches [21].

Conventional empirical approaches tend to exploit the observed features of the echosounder data without modifying them, retaining their physical meaning. This approach allows for direct interpretation of the results based on their physical implications, making it easier for users to understand the analysis outcomes [70]. The relative frequency response [60, 69], illustrated in Figure 2.3, is a well-known method in this stream, leveraging the frequency dependence of the target using multi-frequency echosounders.

The empirical approach can be complemented by trawl sampling [45], a method of collecting biological data on target populations *in vivo* using a trawl net deployed from a vessel. Trawl sampling is typically used for pelagic or semi-pelagic species, such as walleye pollock [71], herring, blue whiting [72], and sandeel [39]. This method is particularly useful for obtaining ground truth annotation of the data on various aspects of the target, such as length, weight,

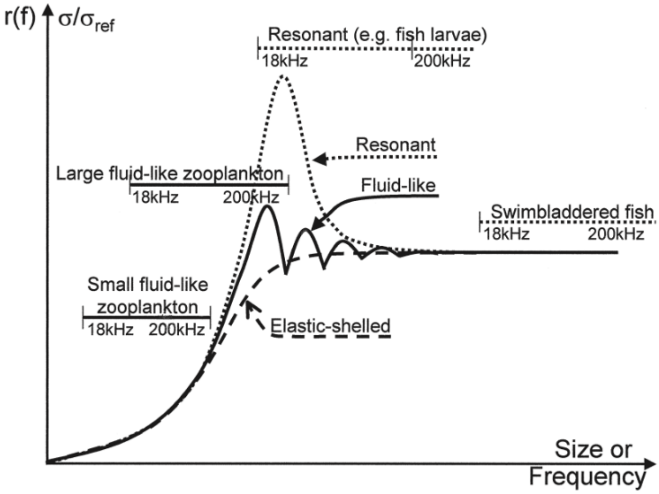


Figure 2.3: A general schematic description of the relative frequency response, r(f). Bands indicate typical positions of selected acoustic categories when measured at frequencies 18–200 kHz. Example adapted from [2].

age, and sex for each individual.

2.2.2 Advanced approach

In the conventional approach,e expert analysts rely on heuristics to manually identify targets in the data [22], with the outcomes of trawl sampling [45] serving as validation for this identification process. This approach can be time-consuming, labor-intensive, and subject to human bias [6, 15, 25], causing the analysis results to depend on the analyst's experience, which makes it difficult to study the behavior and distribution of complex marine ecosystems. Thus, advanced approaches aim to make echosounder data analysis more systematic and automated to achieve consistent results across various studies and applications.

Prior to the introduction of deep learning-based approaches, statistical and pattern recognition methods were applied to the explicit features observed in echosounder data, using techniques such as support vector machines [73, 74], k-nearest neighbors [74], and decision trees [74]. When combined with kernel methods [73, 75, 76], these approaches could achieve improved results by leveraging implicit feature analysis.

Deep learning-based approach Deep learning-based approaches, particularly convolutional neural networks (CNN) [6, 77, 78], have demonstrated a large potential in learning complex and non-linear relationships between observed data and the target variable, such as class annotations. For instance, Brautaset et al. [6] apply U-Net-based semantic segmentation [79] to classify each individual backscattered intensity in echosounder data, resulting in the segmentation map. Other studies [77, 78] also utilize CNNs for predictive image tasks, achieving impressive results. Importantly, the networks employed in these studies [6, 77, 78] are trained in a fully-supervised manner, with annotations provided for all training samples.

Rapid advancements in fully supervised deep learning-based marine environmental monitoring research stem from the integration of various sensing methodologies. These include echosounder data [80], imagery from autonomous underwater vehicles (AUVs), and multi-sensor modalities connected by multiple buoys [81].

As deep learning is combined with data collected through diverse sensing methods, new avenues of research emerge in marine environment monitoring, such as information fusion based on multi-modality [82] and 3D underwater reconstruction [83]. However, most existing methods have been limited to fully-supervised approaches without providing any explainability.

2.3 Data of interest

This thesis presents three papers that examine two different types of marine environment monitoring data. Echosounder data is investigated in Papers I-III, while aerial images of seal pups on sea ice are analyzed in Paper III. A brief description of each of the datasets is presented in Section 2.3.1 and 2.3.2, respectively.

2.3.1 Echosounder data

Sandeel survey The echosounder data used in the presented papers (Papers I, II, and III) is collected during the sandeel survey in the Norwegian North Sea [84]. The sandeel survey aims to investigate the North Sea ecosystem to better understand the distribution, behavior, and ecology of sandeels and their relationship to other marine species. The sandeel (*Ammodytes marinus*) is a small fish that lacks a swim bladder and is known to spend a significant portion of its life burrowing and hiding in sandy seabeds with a low proportion of fine silt and clay particles [85, 86]. During the spring feeding season, adult sandeels emerge from their sandy bottom hiding places at dawn to form large schools in the pelagic upper layer and feed on zooplankton [34]. Sandeels are considered to be a key species in the North Sea ecosystem and are a vital prey species for

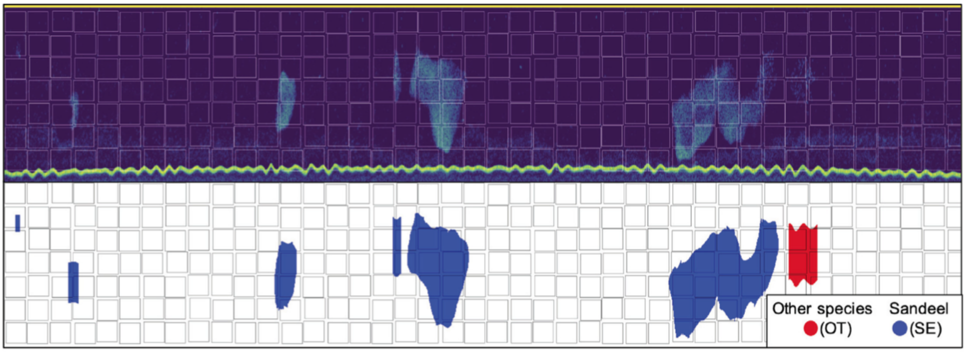


Figure 2.4: Echosounder data (up) and the corresponding pixel-level annotation (down). Image adopted from Paper I.

several predators, including seabirds, seals, and large fish [87], as well as being a significant target for commercial fisheries.

The Norwegian Institute of Marine Research has conducted annual acoustic trawl surveys for sandeel in the northeastern North Sea during April and May since 2005 [34]. The survey series has been conducted using various research vessels equipped with multi-frequency Simrad EK60 echosounder systems with transducers operating at 18, 38, 120, and 200 kHz, except for the year 2012, which utilized a Simrad ME70 sonar to collect 120 kHz data [88]. Since 2014, the vessels have been equipped with a 70 and 333 kHz echosounder. The echosounder systems are calibrated before each survey following standard procedures [48], and during operation, pulse duration and ping repetition frequency are set to 1.024 ms and 3-4 Hz for all frequencies, respectively, while vessel speed is maintained at approximately 10 knots. Echosounder observations are stored as frequency-specific values of the volume backscatter coefficient, s_v, which is a mean backscatter intensity per cubic meter [89].

Data preprocessing The duration of the pulse and ping rate, both related to the time range resolution (horizontal axis), may vary from the standard settings in some instances. To ensure consistency, the data is interpolated onto a time range grid with a resolution of 200 kHz data for all frequency channels to result in a uniform time-range grid of the echosounder data. The 200 kHz data is chosen because it has the highest signal-to-noise ratio of the sandeel species. When a ping is missing, the median ping rate is used. A column of zeros (mapped to -75 dB re m^{-1} after log transformation) is inserted into the missing ping. The seabed is approximated by identifying the depth associated with the highest rate of increase in the vertical gradient for each acoustic ping.

The collected echosounder data is manually annotated based on the frequency response of each school [39] and validated with trawl samples where applicable [45]. The annotation process is done by the same expert analyst across all years with the aid of large-scale survey system (LSSS) post-processing software [69].

The initial target classes consist of sandeel, other species, zero-group sandeel, possible sandeel, and background, where the zero-group sandeel and possible sandeel classes are regarded as minor occurrences that can be disregarded and are merged with the background class. The class possible sandeel denotes instances in which there is a discrepancy between the classification determined by the frequency response and that identified by the expert analyst, with regards to schools of fish suspected to be sandeel. The class zero-group sandeel is introduced for the survey of 2016, to accommodate the atypically high concentration of juvenile specimens.

2.3.2 Aerial images of seal pups on sea ice

The ice-breeding harp and hooded seals are both abundant species in the North Atlantic. There are two geographically separate populations of hooded seals and three of harp seals. These populations have historically been exploited and managed separately. As a result, there is a need to assess their status and monitor changes in abundance in all populations to manage the respective harvests responsibly. Knowledge of seal population sizes is required to estimate the potential interactions of these species with other marine organisms, including commercially important fish species.

In a management framework, precise estimates of key parameters in population models are vital to providing reliable future predictions of the population. To obtain these, independent estimates of pup production using photographic or visual aerial strip transect surveys are used to determine the abundance of harp and hooded seals in the Northwest Atlantic [90], the Greenland Sea [40], and the White Sea [91]. The total abundance is subsequently estimated by fitting a population model to the independent estimates of pup production while incorporating removals and reproductive rates [92]. The number of seal pups is counted either visually along an entire transect (with a known strip width) or from aerial images taken along a transect. A number of parallel transects are surveyed to cover an entire patch of seals. To obtain estimates of total harp or hooded seal populations, several thousand images are typically required [92].

Manual analysis of the photographs is extremely time-consuming and costly, and involves subjective human interpretation by trained experts. The spatial distribution of the seals varies substantially. Typically, the ice-breeding seals will cluster, but due to substantial ice drift, the seals might be scattered over large areas. Often, only a small fraction of the images taken contains seals,

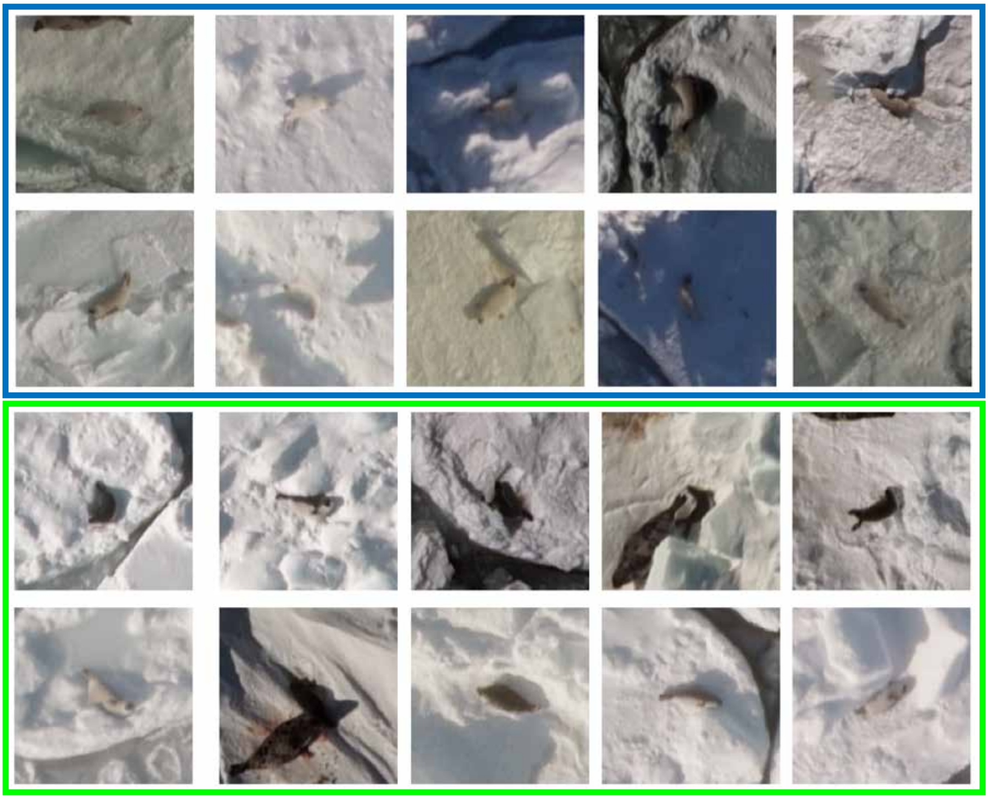


Figure 2.5: Images of seal pups on sea ice. The images in the blue box depict harp seal pups, while those in the green box feature hooded seal pups.

with typically 70-90 percent of harp and hooded seal pup images being empty.

The seal pup dataset consists of aerial photos (RGB) with corresponding annotations indicating the position and species of all seal pups in the images. The aerial photos were acquired during surveys in the West Ice in 2007, 2012, and 2018, and in Canada in 2008, 2012, and 2017. The resolution is about 2 or 3 centimeters, depending on the altitude of the aircraft. The seal pup images used in this study are manually annotated into three classes, namely harp seal, hooded seal, and background. Figure 2.5 shows example images of the seal pups for the two seal classes.

3 | Basic theories of deep learning

Deep learning involves training deep neural networks to recognize patterns and make predictions based on complex, large-scale datasets [24]. Methods based on deep learning extract and identify meaningful features from data, resulting in more accurate and robust predictions. Deep learning has achieved remarkable success in various real-world problems such as medicine [93–95], remote sensing [96–98], and marine science [6, 13, 99, 100], outperforming conventional approaches.

In this chapter, a theoretical background for the deep learning aspects relevant to the thesis is presented. Section 3.1 discusses fundamental topics of deep learning, including gradient descent, the perceptron algorithm [101], and fully connected neural networks (FCNN) [102]. Subsequently, Section 3.2 delves into deep learning-based image tasks relevant to the presented research papers, covering image classification and semantic segmentation, along with their respective backgrounds.

3.1 Introduction to deep learning

3.1.1 Gradient descent

A local minimum of a differentiable function $J(w)$ can be found using the first-order iterative optimization process known as *gradient descent.* The objective is to repeatedly move in the direction opposite to the function's gradient (or approximate gradient) at the current location, since the gradient represents the path of steepest ascent. The mathematical expression for gradient descent is given by:

$$w_{s+1} = w_s - \rho_s \left. \frac{\partial J(w)}{\partial w} \right|_{w=w_s} . \tag{3.1}$$

Here, s represents the current time step in the iteration, while w_s and w_{s+1} denote the parameter of interest at time steps s and $s+1$, respectively. A positive ρ_s determines the impact of the gradient while moving toward a

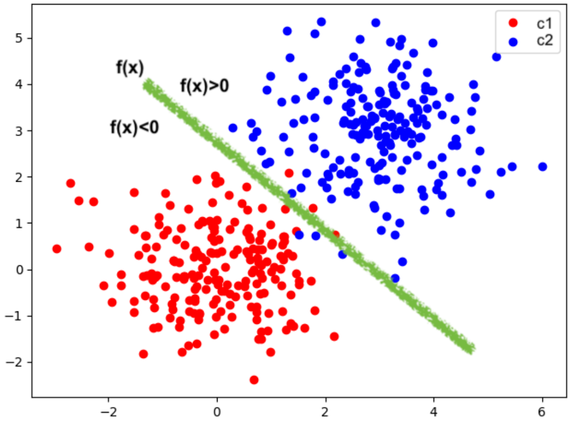


Figure 3.1: Four hundred two-dimensional data points $\mathbf{x}$ and their ground truth classes (w_1 in red and w_2 in blue). The linear discriminant function $f(\mathbf{x})$ aims to classify them into two classes as close to their ground truth as possible.

(local) minimum, also known as the learning rate. It is important to note that the sign of the gradient is negative, as the goal is to move in the opposite direction of the gradient.

Perceptron The perceptron algorithm [101] is one of the earliest data-driven analysis methods that utilizes gradient descent. Figure 3.1 illustrates the mechanism using a binary classification task. To classify the training dataset $\mathbb{X} = \{\mathbf{x}_1 \cdots \mathbf{x}_N\}$, where $\mathbf{x}_i \in \mathbb{R}^2$, a linear discriminant function $f(\mathbf{x})$ is introduced to separate these data points into two classes close to their ground truth classes, denoted by c_1 and c_2, as follows:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b. \tag{3.2}$$

Note that $\mathbf{w} = [w_1, w_2]$, and b represent the learnable parameters for the discriminant function $f(\mathbf{x})$. The discriminant function classifies $\mathbf{x}_i$ to c_1 if $f(\mathbf{x}_i) < 0$. Otherwise, $\mathbf{x}_i$ is classified to c_2. The objective is to determine the parameters $\mathbf{w}$ and b such that the discriminant function $f(\mathbf{x})$ maximizes the accuracy of classifying the dataset $\mathbb{X}$ with respect to their ground truth classes.

By utilizing the linear discriminant function, a cost function $J(\mathbf{w}, b)$ is introduced, which quantifies the degree of misclassification given the current parameters [103], as follows:

$$J(\mathbf{w}, b) = \sum_{\mathbf{x} \in \mathbb{X}_{mis}} \delta_x f(\mathbf{x}) = \sum_{\mathbf{x} \in \mathbb{X}_{mis}} \delta_x(\mathbf{w}^T\mathbf{x} + b). \tag{3.3}$$

In this equation, $\mathbb{X}_{mis} \subset \mathbb{X}$ denotes the misclassified data points. δ_x ensures the cost function to be non-negative, as δ_x shares the same sign as $f(\mathbf{x}_i)$. For instance, $\delta_x = 1$ if $\mathbf{x}_i$, belonging to c_1, is misclassified to c_2 due to $f(\mathbf{x}_i) > 0$. Conversely, $\delta_x = -1$ if $\mathbf{x}_i$, belonging to c_2, is misclassified to c_1 when $f(\mathbf{x}_i) < 0$.

The cost function $J(\mathbf{w}, b)$ can be minimized using gradient descent, as presented in Equation 3.1. Assuming that the minimum cost is achieved at timestep s^* and that the learning rate ρ_s in Equation 3.1 is a fixed value (e.g., $\rho_s = \rho$), analytic solutions for the learnable parameters $\mathbf{w}_{s^*}$ and b_{s^*} can be derived as follows:

$$\begin{aligned} \mathbf{w}_{s^*} &= \mathbf{w}_{s^*-1} - \rho\frac{\partial J(\mathbf{w}, b)}{\partial \mathbf{w}} = \mathbf{w}_{s^*-1} - \rho \sum_{\mathbf{x}\in\mathbb{X}_{mis}} \delta_x \mathbf{x} \\ b_{s^*} &= b_{s^*-1} - \rho\frac{\partial J(\mathbf{w}, b)}{\partial b} = b_{s*-1} - \rho \sum_{\mathbf{x}\in\mathbb{X}_{mis}} \delta_x \end{aligned}. \tag{3.4}$$

3.1.2 Cost function

The perceptron algorithm is suitable for linear classification problems only. As such, it is often viewed as outdated for more complex real-world problems. Nonetheless, the core concept of the perceptron algorithm persists and forms the theoretical foundation of deep neural networks. By illustrating the perceptron algorithm, we clarify the definition of the cost function, which remains applicable to deep neural networks.

Based on δ_x in Equation 3.3, a function $\sigma(z)$ is defined as:

$$\sigma(z) = \begin{cases} z, & \text{if } z > 0 \\ -z, & \text{if } z < 0 \end{cases}. \tag{3.5}$$

Using $\sigma(z)$, the cost function $J(\mathbf{w}, b)$ can be rewritten as:

$$J(\mathbf{w}, b) = \sum_{\mathbf{x}\in\mathbb{X}_{mis}} \sigma(\mathbf{w}^T\mathbf{x} + b). \tag{3.6}$$

In this equation, $\sigma(z)$ plays an essential role in ensuring the cost function remains positive, e.g., $J(\mathbf{w}, b) \geq 0$. The minima, e.g., $J(\mathbf{w}, b) = 0$, can be achieved when all the training samples $\mathbf{x}$ are correctly classified to their ground truth classes, e.g., $\mathbb{X}_{mis} = \{\emptyset\}$.

However, reaching perfect classification accuracy is often challenging in real-world data. Thus, the cost function is rewritten to include the entire training samples instead of only the misclassified ones while keeping the cost

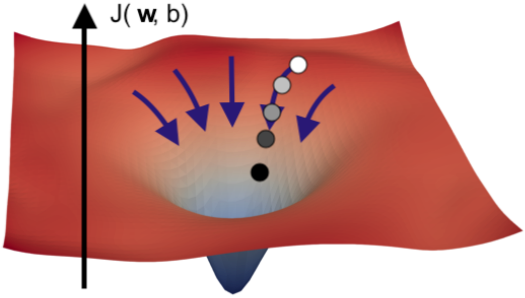


Figure 3.2: An example of the cost function in the weight-bias space. Image inspired from [3].

function positive:

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} L\Big(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i\Big). \tag{3.7}$$

In this equation, $\mathbf{x}_i$ represents the training input sample, y_i takes the ground truth value of the corresponding input sample, N is the number of training samples, $f_{\boldsymbol{\theta}}(\mathbf{x}_i)$ is the model prediction of the input $\mathbf{x}_i$, and $L(\cdot,\cdot)$ is the sample-wise loss, which is non-negative. The choice of $L(\cdot,\cdot)$ depends on the task, which will be described later. It is important to note that this approach is referred to as *fully-supervised* learning when each input example $\mathbf{x}_i$ is annotated by the ground truth y_i [24].

Figure 3.2 provides a visualization of an example cost function $J(\mathbf{w}, b)$ in the weight-bias space, where $\mathbf{w}$ and b represent the learnable parameters $\boldsymbol{\theta}$ in Equation 3.7. Each ball signifies the parameters at a specific time step in the sequence (progressing from white to black), and the ball's trajectory demonstrates the learning process towards the (local) minima based on gradient descent. It is worth mentioning that not all cost functions demonstrate the same level of smoothness as the one depicted in Figure 3.2.

Activation function In the cost function in Equation 3.7, the discriminant function $f_{\boldsymbol{\theta}}(\mathbf{x})$ is not differentiable since the first order derivative of $\sigma(z)$ is not defined at $z = 0$. To apply gradient descent to the cost function $J(\boldsymbol{\theta})$, the function $\sigma(z)$ needs to be adjusted with a differentiable function. Additionally, to compute the sample-wise loss $L\big(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i\big)$ on a manageable scale, it is necessary for the model predictions to belong to a certain range, along with the ground truth values y_i.

Therefore, an activation function $\sigma(z)$ is defined to meet the aforementioned requirements. One commonly used activation function in the perceptron algorithm is the non-linear hyperbolic tangent function, denoted as

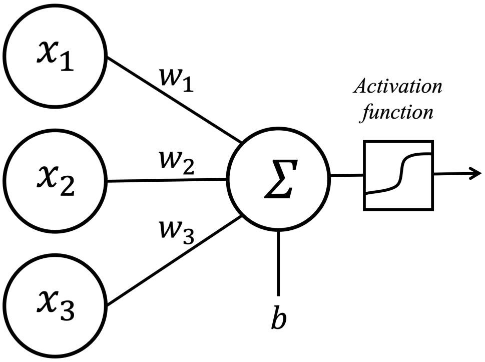


Figure 3.3: A neuron, the smallest computation unit for neural networks, which models $f_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b)$. The perceptron algorithm is based on a single neuron unit.

$\sigma(z) = tanh(z)$. The *tanh* function is differentiable and has an analytic solution for the first-order derivative, making it suitable for gradient descent. Its output range is between -1 and 1, which aligns with the ground truth value $y_i \in \{-1, 1\}$, where $\mathbf{x}_i \in c_1$ if $y_i = -1$ and $\mathbf{x}_i \in c_2$ otherwise. Figure 3.3 shows a schematic of the perceptron algorithm, which is also known as a neuron.

By utilizing a non-linear activation function, it becomes possible to analyze non-linear relationships in data, while also facilitating the generalization and adaptation of a wide range of input types. Moreover, incorporating non-linear activation functions when stacking multiple layers of neurons enables the network to more effectively handle and process intricate data patterns.

3.1.3 Fully connected neural networks

Building upon the concepts of neurons and non-linear activation functions, a fully-connected neural network (FCNN) is proposed [102]. The FCNN achieves a higher level of computational capacity by utilizing stacked layers of neurons that are fully connected between adjacent layers, resulting in improved performance on a wide range of tasks [24]. FCNN is also referred to as a multi-layer perceptron (MLP) due to its stacked architecture. In the multi-layer architecture, data is passed through each layer, with the output of one layer serving as the input for the next. An example architecture of FCNN with three layers of nodes, including an input layer, a hidden layer, and an output layer, is depicted in Figure 3.4.

The training of the network entails the minimization of the cost function $J(\boldsymbol{\theta})$ in Equation 3.7, where the required network predictions $f_{\boldsymbol{\theta}}(\mathbf{x})$ are defined in the output layer of the network. During training, the parameters are iteratively updated towards their optimal values. This involves computing the

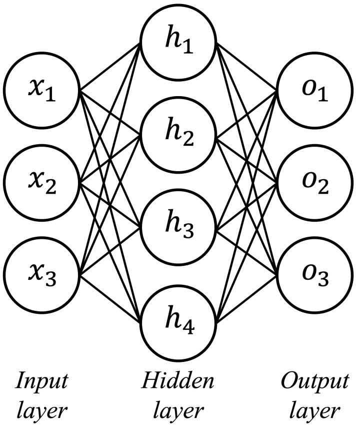


Figure 3.4: The architecture of fully connected neural networks with a single hidden layer.

gradient at each step, which represents the first-order derivative of the cost function with respect to the parameter at the current time step. The backpropagation algorithm is used to compute the gradient of the parameters in the input layer using the cost function defined in the output layer, as will be discussed in the next section. Finally, all parameters are updated simultaneously using the gradient descent update rule presented in Equation 3.1.

This is also referred to as end-to-end learning [24]. The whole neural network is trained together, from input to output, without requiring handcrafted features or intermediate representations. The focus is on optimizing a single objective function that reflects the overall performance of the system.

Backpropagation When considering a FCNN, it is important to note that each parameter in the FCNN has its analytic solution for the gradient with respect to the cost function $J(\boldsymbol{\theta})$. The process of obtaining this solution is referred to as backpropagation. The main idea behind backpropagation is to recursively apply the chain rule of calculus to compute the gradients [24]. During the forward pass, the inputs are fed through the network and the output is computed. During the backward pass, the gradients of the loss function with respect to the output are first computed. Then, these gradients are propagated backwards through the network to compute the gradients of the loss function with respect to the parameters in each layer.

Taking the architecture presented in Figure 3.4 as an example, the output layer $\mathbf{o}$ and the hidden layer $\mathbf{h}$ are respectively denoted by:

$$\begin{aligned} \mathbf{o} = q_{\boldsymbol{\theta}}(\mathbf{h}) = \mathbf{W}_q^T\mathbf{h} \\ \mathbf{h} = p_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_p^T\mathbf{x} \end{aligned} . \tag{3.8}$$

In this equation, $q_{\boldsymbol{\theta}}(\cdot)$ and $p_{\boldsymbol{\theta}}(\cdot)$ represent the layers of neurons situated between the hidden and output layers, and the input and the hidden layers, respectively. The parameters in each layer are represented by $\mathbf{W}_q$ and $\mathbf{W}_p$.

The gradients of the parameters in the $q_{\boldsymbol{\theta}}(\cdot)$ layer can be defined based on the chain rule. The analytic solution is calculated as follows:

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \mathbf{W}_q} = \frac{\partial J(\boldsymbol{\theta})}{\partial \mathbf{o}} \frac{\partial \mathbf{o}}{\partial \mathbf{W}_q} = \frac{\partial J(\boldsymbol{\theta})}{\partial \mathbf{o}} \mathbf{h}. \tag{3.9}$$

In this equation, $J(\boldsymbol{\theta})/\partial \mathbf{o}$ is determined by the choice of the sample-wise loss $L(\cdot,\cdot)$. Another partial derivative term, $\partial \mathbf{o}/\partial \mathbf{W}q$, can be simplified to $\mathbf{h}$ by utilizing the relationship $\mathbf{o} = \mathbf{W}q^T\mathbf{h}$ from Equation 3.8.

Analogously, the analytic solution for the gradients of $\mathbf{W}_p$ are defined as follows:

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \mathbf{W}_p} = \frac{\partial J(\boldsymbol{\theta})}{\partial \mathbf{o}} \frac{\partial \mathbf{o}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{W}_p} = \frac{\partial J(\boldsymbol{\theta})}{\partial \mathbf{o}} \mathbf{W}_q \mathbf{x}. \tag{3.10}$$

3.1.4 Challenges

While backpropagation is a powerful tool for training neural networks, it is not without its challenges. We will discuss some of the common issues and limitations of backpropagation, and how researchers have attempted to address them.

Vanishing and exploding gradient Backpropagation enables the generalized training of neural networks regardless of architecture. However, in practice, it presents challenges in propagating gradients deeply into the network. Backpropagation relies on the multiplication of the partial derivative terms multiplied together, as shown in Equations 3.9 and 3.10, If some partial terms are close to zero, the gradient can vanish during backpropagation. In such cases, backpropagation is cut off in the middle of the network, and the parameters in the layers near the input may not be updated, remaining unchanged due to the zero gradient.

Conversely, if some partial terms are much greater than one, the gradient can exponentially increase, making the learning process unstable. In such cases, the network may fail to converge to the minimum point of the cost function $J(\boldsymbol{\theta})$.

Overfitting In addition to gradient-related issues, overfitting is a common problem observed during network training. Overfitting occurs when the network fails to generalize to unseen data, such as the test set, due to overemphasizing details in the training examples. To avoid overfitting, regularization techniques, such as early stopping, dropout, or weight decay, can be applied while monitoring the learning curves of both the training and test sets during iterations [104].

3.1.5 Remedies

To address the gradient-related issues and overfitting problem, various techniques have been proposed in the literature [24]. These techniques aim to stabilize the learning process and improve the generalization ability of the network.

Rectified linear unit The choice of an activation function is crucial in avoiding the vanishing gradient problem. Although non-linear activation functions, such as the sigmoid function, are employed to add non-linearity to neural networks, a significant issue with such functions is the two-sided gradient saturation, which may limit the range of values that the gradient can backpropagate.

To address this problem, the rectified linear unit (ReLU) has been introduced as an activation function [105]. This partially linear function mitigates the two-sided gradient saturation problem by making it one-sided, allowing the gradient to backpropagate without saturation on the open side.

Equation 3.11 illustrates ReLU [105] and its gradient, respectively:

$$\begin{aligned} \sigma(z) &= \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{otherwise} \end{cases} \\ \frac{\partial \sigma(z)}{\partial z} &= \begin{cases} 1, & \text{if } z > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} . \tag{3.11}$$

In this equation, $\sigma(z)$ and $\partial\sigma(z)/\partial z$ are the ReLU function and the gradient, where the gradient is not saturated in the open side for $z > 0$.

Input standardization Before processing the input layer $\mathbf{x}$ into the network, it is recommended to standardize the input to maintain a consistent scale among features and enhance the generalization capabilities of the model to unseen data [24]. However, it should be noted that domain knowledge is required for data preprocessing since the numerical range of $\mathbf{x}$ of interest may vary based on the characteristics of the data and the analysis purpose. To standardize the input, we can use either min-max scaling or Z-score normalization, which are illustrated below in Equation 3.12:

$$\begin{aligned}\mathbf{x}_{new} &= \frac{\mathbf{x} - x_{min}}{x_{max} - x_{min}} &&: \text{min-max scaling}\\ \mathbf{x}_{new} &= \frac{\mathbf{x} - E[\mathbf{x}]}{std[\mathbf{x}]} &&: \text{Z-score normalization}\end{aligned}. \tag{3.12}$$

Batch normalization Batchnorm (BN) [106] is a commonly used method to ensure that feature representations have values that the neural network can work with. This technique involves introducing learnable parameters at a mini batch level, which helps re-center and re-scale the hidden layer. In the training phase, each element of the hidden layer is transformed to have a normal distribution based on the mean and variance of the mini batch. The learnable parameters are then used to re-scale and re-center the normalized element, with the gradient descent algorithm updating these parameters.

Equation 3.13 provides a detailed illustration of how the BN layer behaves during training:

$$\begin{aligned}h_{BN}^{(d)} &= \frac{h^{(d)} - \mu^{(d)}}{\sqrt{\sigma^{2(d)} + \epsilon}}\\ h_{out}^{(d)} &= \gamma^{(d)} h_{BN}^{(d)} + \beta^{(d)}\end{aligned}, \tag{3.13}$$

where $h^{(d)}$ indicates the d^{th} element of $\mathbf{h}$, $\mu^{(d)}$ and $\sigma^{2(d)}$ represent the mean and variance of $h^{(d)}$ in the mini batch, respectively, and $\gamma^{(d)}$ and $\beta^{(d)}$ the learnable parameters involving in re-scaling and re-centering, respectively. The constant ϵ is added to the denominator to avoid division by zero.

In the inference phase, the BN layer uses the averaged batch mean and variance over the batches in the training set, e.g., $E[\mu_{TR}^{(d)}]$ and $\frac{M}{M-1}E[\sigma_{TR}^{2(d)}]$, instead of using the mean and variance in the current mini batch in the test set.

Regularization Although regularization is not directly associated with the vanishing gradient problem, it is an essential technique in deep learning that prevents overfitting and manages model complexity. As a result, regularization techniques contribute to achieving simpler models that can learn global and general feature representations.

Regularization techniques are divided into two groups, depending on whether they explicitly impact neural activation or not. The first group includes regularizing layers, such as Batchnorm [106] and Dropout [107], and regularizing loss terms, such as L2 or L1 norms [24]. Batchnorm smooths the optimization landscape using re-scaling and re-centering, making the gradients more predictive and stable, which leads to faster training [108]. Dropout temporarily disables some neuron units in the network during training, reducing co-adaptation among the units and learning sparser feature representations. In addition, the network connection varies every iteration, making the learning process equivalent to the ensemble of the sub-networks, which improves generalization.

The regularizing loss is a mathematical term that is included in the cost function $J(\boldsymbol{\theta})$ and optimized by gradient descent. Commonly applied regularizing losses are defined by learnable parameters, denoted as $\Omega(\boldsymbol{\theta})$. The aim of the regularization loss is to limit the capacity of the network by penalizing parameters that become too large, which reduces the risk of overfitting. The impact of the regularization loss can be adjusted by a coefficient α, where the regularized cost function is given by $J(\boldsymbol{\theta}) + \alpha\Omega(\boldsymbol{\theta})$. The L_2 and L_1 norms are often chosen based on the analysis purpose, where the L_2 norm is the square root of the sum of squares of the parameters, while the L_1 norm is the sum of the absolute values of the parameters. The choice of the norm depends on the desired regularization effect, as the L_2 norm encourages small weights overall, while the L_1 norm promotes sparse weights, where only a few parameters are significantly different from zero.

The second group includes regularization techniques not explicitly related to neural activation, but that mitigate learning challenges in neural networks. Early stopping [109] is used to avoid overfitting by stopping iterative training when the current parameters fail to address the test set as much as they do the training set. Data augmentation [24] is a technique that expands the training data by including slightly altered samples originating from the training data. This technique is often used in image analysis, where a set of augmentation tools, such as flipping, cropping, shifting, rotating, scaling, and re-coloring, are applied. Data augmentation has been recently re-spotlighted due to its use in self-supervised learning methods [110–113] that leverage data augmentation to learn more fundamental and discriminative image feature representations.

3.2 Image tasks based on deep learning

This section explores image tasks based on deep learning, with a particular focus on convolutional neural networks (CNNs), which are highly effective for structured data, such as images [24]. The research papers in this thesis use CNNs extensively, owing to the structured nature of the multi-frequency echosounder data, which has water depth and sailing time represented on the vertical and horizontal axes, respectively.

After providing an overview of CNNs, this section discusses various image

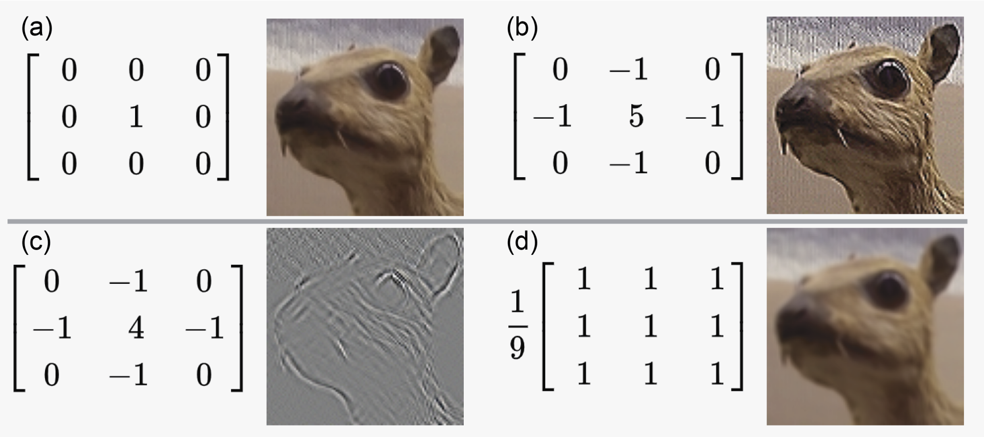

(a)
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
(b)
$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
(c)
$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
(d)
$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Figure 3.5: Outputs of an image after being processed by different convolutional filters: (a) Identity filter, (b) Sharpen filter, (c) Edge detection filter, and (d) Box blur filter.

analysis tasks studied in the accompanying papers, including image classification and semantic segmentation.

3.2.1 Convolutional neural networks

CNNs are a popular deep learning framework specifically designed for processing structured data, such as images and videos [24]. A typical CNN architecture consists of multiple layers of neural networks, including convolutional and pooling layers, arranged hierarchically to extract unique features from the input data.

Convolutional layers extract features from the input by applying convolution operations. Each convolutional layer contains multiple filters that slide over the input to capture spatial features, such as edges or corners. In Figure 3.5, the outputs of an image after being processed by different convolutional filters are illustrated, showing how each filter is designed to detect specific features in the image, allowing the network to extract unique features and patterns for image analysis tasks.

These convolutional layers are locally connected, indicating that each filter in a convolutional layer is only connected to a small region of the input data. This local connectivity allows for efficient computation and reduced memory usage since the network only processes a small portion of the input at a time, in contrast to fully connected neural networks. These convolutional layers share the same set of weights across different regions of the input data, allowing the network to detect the same features at different positions in the input. This property, referred to as parameter sharing, contributes to the robustness of

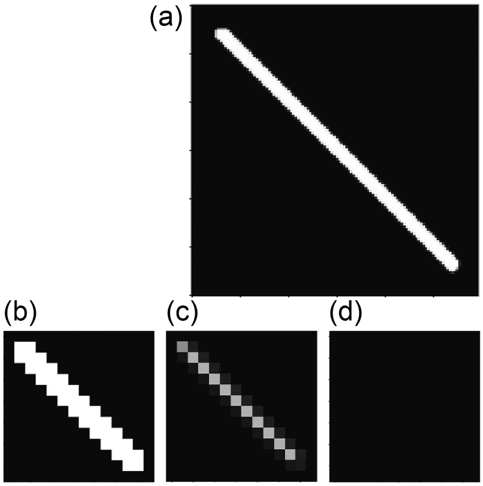


Figure 3.6: Outputs of an image being processed by different pooling layers with a filter size of 8x8 and a stride of 8: (a) Original image, (b) Max pooling, (c) Average pooling, (d) Min pooling.

CNNs by providing translation equivariance [114].

Pooling layers are typically used in CNNs to downsample the feature maps generated by the convolutional layers, reducing the spatial dimensions of the input and extracting the most important information from the feature maps. This process helps to improve the computational efficiency of the network and also serves to make the network more robust to small variations in the input.

By reducing the spatial dimensions of the input, pooling layers facilitate the learning of abstract features in CNNs. This is because abstract features tend to be invariant to small changes in the input, and by downsampling the feature maps, the pooling layers help the network to identify these invariant features at different scales and locations in the input. This ability to learn abstract features is one of the key strengths of CNNs and is what makes them particularly effective for image tasks.

Figure 3.6 shows the outputs of an image being processed by different pooling layers. Each pooling operation reduces the spatial size of the input feature maps by aggregating adjacent values, resulting in a more compact representation, while different pooling mechanisms result in different feature maps.

The hierarchical feature extraction design of CNNs enables it to extract complex features from the input data in a deep and hierarchical manner, allowing for better representation and understanding of the underlying data. Coupling these CNN features with the advantages of neural networks, including end-to-end learning as discussed in Section 3.1, enables CNN to efficiently learn to analyze structured data.

3.2.2 Image classification

Image classification is a classical image analysis task, which involves classifying an image based on its dominant object or objects [103]. In recent years, computer vision researchers have extensively studied this task with the aim of automating the classification process.

Traditional image classification is typically performed in two steps: feature extraction and feature classification. In the feature extraction step, visual features are extracted from the input image, by transforming the raw pixel data into a set of features that can be easily processed by machine learning algorithms. This is a critical step since the quality of the extracted features will impact the accuracy of the classification. Commonly used techniques for feature extraction include scale-invariant feature transform (SIFT) [115] and gradient location orientation histogram (GLOH) [116]. In the feature classification step, the extracted features are related to specific classes. To classify a new image based on its features, machine learning algorithms, including support vector machines (SVMs) [117] or random forests [118], are employed.

While the traditional approach to image classification has proven effective in some cases [119, 120], its classification performance can be limited by the quality of the extracted features as each analysis step operates according to its own computational logic. Moreover, the need for significant human intervention in the computational logics mentioned above makes it challenging to develop automated processes for handling large amounts of data.

CNN-based image classification CNNs have emerged as a promising approach to address the limitations of traditional image classification techniques [24]. One key advantage of CNNs is the ability to feature extraction and classification into a single, unified network, enabling the network to extract hierarchical and abstract features more efficiently from large and complex image data.

A typical CNN architecture for image classification consists of two primary components, namely the feature extractor and the feature classifier. These two components are connected in series and simultaneously trained using backpropagation, allowing the CNN to learn both basic visual features, such as edges and corners, as well as more complex high-level features like shapes, patterns, and objects at each iteration.

The feature extractor consists of multiple convolutional and pooling layers, which are hierarchically arranged to extract essential features from the input data while reducing their dimensionality [24]. The resulting feature maps are

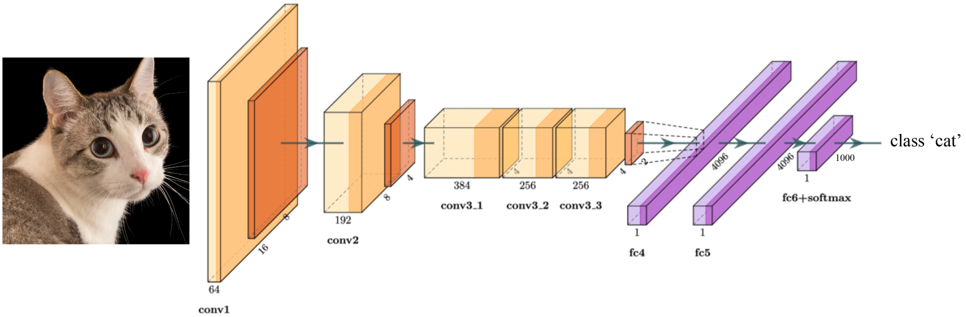


Figure 3.7: The network architecture of AlexNet [4].

then flattened and passed to the feature classifier, which is typically implemented using a fully connected neural network (FCNN). The role of the feature classifier is to take the flattened feature vector and produce a probability distribution over the possible classes. The softmax function is often used as the output activation function to ensure that the predicted probabilities sum to one.

The selection of an appropriate CNN architecture for image classification is dependent on several factors, including the complexity of the classification task, the quantity of available data, and the computational resources available [103]. An effective starting point in selecting a network architecture is to examine the architecture of existing CNNs that have been successful in similar classification tasks. Popular CNN architectures that have shown remarkable performance in image classification tasks include AlexNet [4], VGGs [121], and ResNet [122].

Figure 3.7 shows AlexNet [4] as an example of a CNN architecture, where the feature extractor and feature classifier are highlighted in different colors. AlexNet [4], an eight-layer CNN, is the milestone for deep learning based image classification, which defeated the runner-up by a large margin in the 2012 ImageNet challenge. Since then, CNN-based approaches have become a primary choice for image classification in various domains, such as classification of fish species [123], microscopic foraminifera [124], noctilucent cloud [125], and the northern lights [126], to name a few.

3.2.3 Semantic segmentation

Semantic segmentation, a crucial image processing task, has emerged as a vital technique for understanding and interpreting complex scenes in images and videos. It involves partitioning an image into distinct regions, each corresponding to a specific class or object, enabling machines to comprehend the

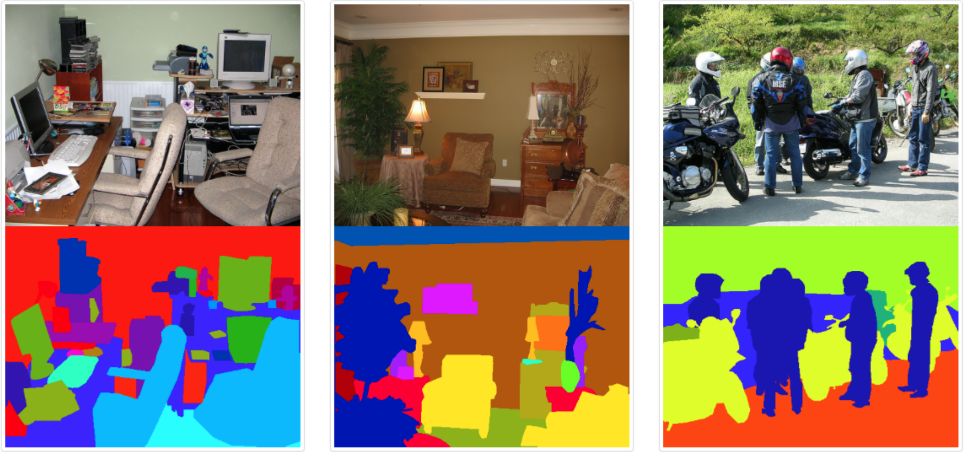


Figure 3.8: Sample images (top row) and their corresponding annotated ground-truth segmentation maps (bottom row) from the PASCAL VOC 2010 dataset [5].

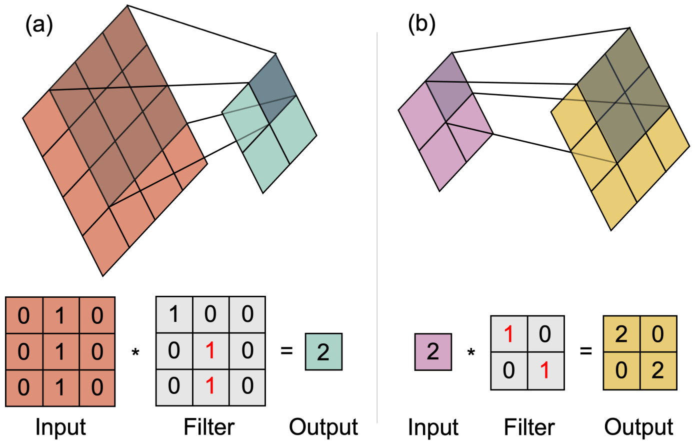
visual world in a manner akin to human perception. Traditionally, it has been considered a challenging task due to factors such as large distribution variance and significant class imbalance among objects in the input data [127].

The emergence of CNNs has accelerated advancements in semantic segmentation. These advancements can be attributed to the inherent advantages of CNNs. As discussed in previous sections, these advantages include learning hierarchical feature representations, robustness to translations, end-to-end training capabilities, and efficiency in training on large-scale datasets [128].

Leveraging these enhanced capabilities, semantic segmentation now finds application in a variety of domains, for instance, self-driving vehicles [129], medical imaging for polyp detection and tumor segmentation [93, 130–132], land cover classification [133, 134], and change detection [135, 136] in earth observation. Figure 3.8 illustrates the diversity and complexity of object categories and scenes within the PASCAL VOC 2010 dataset.

This section concentrates on topics relevant to the included papers. The section begins with an overview of the network architecture used for semantic segmentation, followed by a discussion of the U-Net architecture [79]. The section also includes an illustration of the transpose convolutional layer [137] that enables upscaling in the U-Net architecture, followed by a discussion of the various loss functions that can be used for semantic segmentation.

Architecture CNN-based semantic segmentation methods typically employ an encoder-decoder network architecture [138]. The encoder serves a role similar to the feature extractor of the CNN for image classification. It extracts



$$\begin{bmatrix}0&1&0\\0&1&0\\0&1&0\end{bmatrix} * \begin{bmatrix}1&0&0\\0&1&0\\0&1&0\end{bmatrix} = \begin{bmatrix}2\end{bmatrix}$$

$$\begin{bmatrix}2\end{bmatrix} * \begin{bmatrix}1&0\\0&1\end{bmatrix} = \begin{bmatrix}2&0\\0&2\end{bmatrix}$$

Figure 3.9: Comparison of the mechanisms for (a) a 3x3 convolutional filter and (b) a 2x2 transpose convolutional filter, both with a stride of one.

features for the decoder.

The decoder receives the extracted features as input and reconstructs an output that matches the input size. To compensate for the reduced dimensionality caused by the pooling layers in the encoder, the decoder incorporates upscaling layers, which increase the dimension of the feature map, in the network architecture [24]. Non-parametric upscaling layers include linear interpolation and nearest neighbor, while parametric approaches are also available by, for instance, leveraging layers such as transpose convolutional layers [137].

At the end of the decoder, the final layer, also known as the segmentation head, is applied. A softmax function is usually included in the layer to assign class probabilities to each pixel in the final feature map. The class with the highest probability for each pixel determines the predicted segmentation map.

Prominent CNN architectures for semantic segmentation include fully convolutional networks (FCN) [139] and U-Net [79]. FCN replaces the fully connected layers of traditional CNN with convolutional layers, making them capable of handling input images of any size with learnable upscaling. U-Net [79] will be discussed in greater detail later.

Transpose convolutional layer Transpose convolutional layers [137], also known as deconvolutional layers or fractionally strided convolutional layers, are a type of upscaling layer employed in CNN for semantic segmentation. They serve as the inverse operation of a standard convolutional layer, effectively reversing the process of spatial downsampling. In contrast to standard convolutional layers, transpose convolutional layers have a different function.

They take a small input feature map and apply learnable filters to produce a feature map with larger spatial dimensions. Figure 3.9 provides a visual comparison of the mechanisms behind (a) a convolutional filter and (b) a transpose convolutional filter, demonstrating the differences in their operations.

U-Net U-Net was originally designed for biomedical image segmentation [79] but has since been applied to various domains, including marine environment monitoring [6, 15]. Its U-shaped architecture consists of multiple computational stages in both the encoder and decoder, with corresponding stages in each component connected through skip connections. Figure 3.10 illustrates an example of the U-Net architecture [6].

The encoder is responsible for extracting features from the input image [138]. It consists of a series of convolutional layers, each followed by a BN layer (if needed) and a ReLU activation function, and a max-pooling layer for downsampling. This process is repeated across multiple stages, with each stage gradually reducing the image size while capturing increasingly complex features. The extracted features at each stage are then sent to the corresponding stage in the decoder through a skip connection.

One of the notable properties of U-Net is the skip connections, which link the corresponding stages in the encoder and decoder. These connections enable the decoder to fuse features of different complexity levels. As a result, the network retains finer details from the original image, producing more accurate segmentation maps.

The decoder restores the spatial resolution lost during the pooling process [138]. It consists of upscaling layers (transpose convolutional layers), which expand the feature maps back to the original input size. After each upscaling stage, the feature map is concatenated with the corresponding feature map arrived by the skip connections, providing high-resolution details to the expanding feature maps. A series of a convolutional layer, a BN layer (if needed), and a ReLU activation follows after the transpose convolution layer.

The final layer of U-Net is a 1x1 convolutional layer, followed by a softmax function, which assigns class probabilities to each pixel in the output feature map. The class with the highest probability for each pixel determines the predicted segmentation.

Loss function The selection of a loss function is crucial, as the learning behavior of the network depends on this choice [138]. Common choices for semantic segmentation include cross entropy loss [140], focal loss [141], and dice loss [142, 143].

Cross entropy loss (CE) [140] is a widely applied loss function not only in

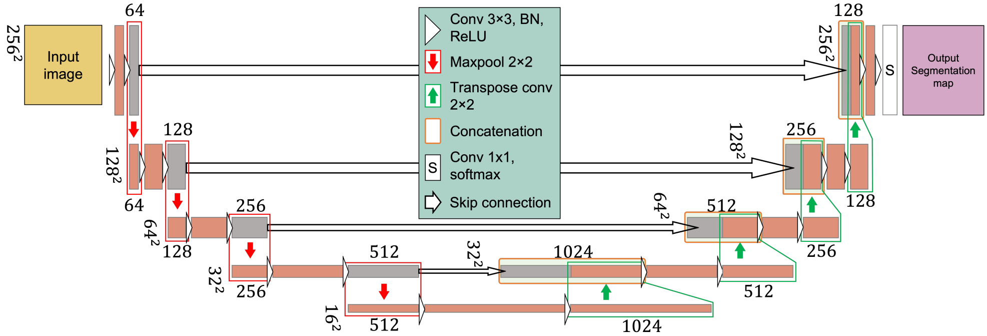


Figure 3.10: An example U-Net architecture used in segmenting echosounder data [6].

semantic segmentation [6, 15, 93] but also in image classification [25, 144]. CE measures the difference between two probability distributions [138], making it suitable for multi-class scenarios with a softmax output and a one-hot encoded label.

In a multi-class scenario, the cross entropy loss, L_{CE}, is defined as:

$$L_{CE} = -\log \hat{y}, \tag{3.14}$$

where $\hat{y}$ is a softmax output of the class to which the ground truth annotation belongs.

Focal loss [141] is an advanced version of CE, designed for extreme class imbalance scenarios. It helps prevent the model from being biased towards the majority classes and performing poorly on the minority classes. The focal loss addresses this issue by reducing the contribution of easy examples during training. This allows the model to focus more on learning difficult examples and minority classes, leading to better overall performance [138]. The focal loss L_F is defined as:

$$L_F = -\log \hat{y}(1-\hat{y})^{\gamma}. \tag{3.15}$$

In this equation, $\gamma \geq 0$ is a focusing hyperparameter that regulates the down-weighting of easy examples. A higher value of γ will result in a stronger focus on hard examples, while a lower value will make the model focus more evenly on all examples.

Dice loss [142] is a generalization of the Dice coefficient [143], which measures the degree of overlap between the predicted and ground truth segmentation by comparing the size of their intersection to the average size of the two sets. This differs from the losses mentioned above, as it does not rely on probability distributions. The Dice loss L_D is defined as:

$$L_D = \frac{1-\hat{y}}{1+\hat{y}}. \tag{3.16}$$

There are other available choices for loss functions for semantic segmentation, including Tversky loss [145], sensitivity specificity loss [146], and Hausdorff Distance [147], to name a few.

In conclusion, selecting an appropriate loss function is essential for achieving optimal performance in semantic segmentation tasks. By understanding the unique characteristics and advantages of various loss functions, researchers can make informed decisions to tailor their models to specific challenges and improve their segmentation results.

4 Advanced deep learning theories

Although deep learning has made significant strides in processing complex and high-dimensional data, the field continues to face several notable challenges [148]. This thesis aims to address some of these challenges, including the limited availability of labeled data and the lack of explainability, by proposing novel deep learning methods.

Specifically, Papers I and II address the limited availability of labeled data by proposing novel semi-supervised methods for image tasks. Paper I presents a semi-supervised image classification method that integrates the underlying data structure with a limited amount of annotated data within a single network. Paper II builds upon the idea from Paper I to propose a semi-supervised semantic segmentation method.

Meanwhile, Paper III aims to address the lack of explainability and proposes a novel explainable deep learning method based on the information bottleneck framework [36, 149]. The proposed method formulates essential requirements to the objective function that the explanation should learn to encompass, such as sufficiency, minimality, and interactivity of the explanation.

This chapter aims to provide an overview of advanced deep learning theories, presenting the theoretical foundation of the three included papers that form the core of this thesis. Section 4.1 discusses the theoretical background of semi-supervised deep learning, focusing on how both labeled and unlabeled portions of data contribute to training a single network. Section 4.2 elaborates on the explainability of deep neural networks, with an emphasis on the application of the information bottleneck framework.

4.1 Semi-supervised deep learning

Semi-supervised deep learning is a learning scheme for neural networks, striving to fully exploit datasets that contain both labeled and unlabeled data. In real-world situations where acquiring labeled data is challenging or costly [150], semi-supervised deep learning proves especially beneficial [151–153], as it can

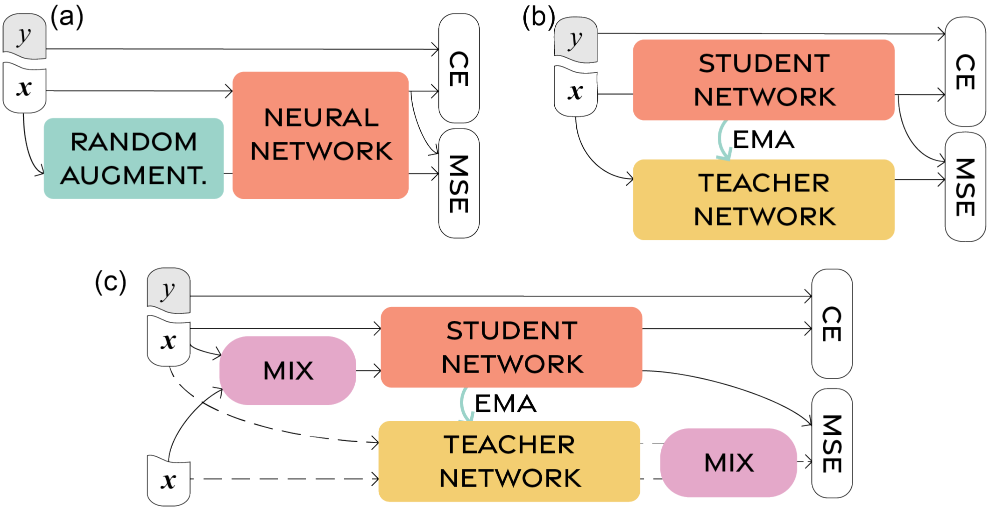


Figure 4.1: Overview of a few consistency training methods. (a) Π model, (b) mean teacher, (c) interpolation consistency training.

leverage the extensive amounts of available unlabeled data to enhance analysis performance.

This section focuses on two primary approaches in semi-supervised deep learning. Section 4.1.1 explores consistency training [154], which is based on the principle of the smoothness assumption [155]. This principle posits that if a small and realistic perturbation is applied to an input, the model's prediction should not change significantly [156]. Meanwhile, Section 4.1.2 investigates pseudo-labeling, an approach that assigns labels to unlabeled data using a model trained on labeled data. The model's predictions for the unlabeled data act as pseudo-labels, allowing the integration of unlabeled data into the training process to enhance performance. The relevance of these approaches to the presented papers is further discussed in Section 4.1.3.

4.1.1 Consistency training

Consistency training seeks to ensure that a network produces similar outputs for perturbed versions of the same input [154, 156]. By doing so, it aims to push the decision boundary into lower-density regions, leading to better class separation. A common approach to achieve this is incorporating a consistency loss into the objective function [151, 152, 157]. Therefore, the objective function often consists of a supervised loss $\mathcal{L}_s$, a consistency loss $\mathcal{L}_u$, and a weight w to balance learning from both losses:

$$\mathcal{L} = w\mathcal{L}_u + \mathcal{L}_s. \tag{4.1}$$

This section discusses three consistency training approaches, including the Π model [157], mean teacher [152], and interpolation consistency training [151]. An overview of these methods is illustrated in Figure 4.1.

Π model With an input $\mathbf{x}$ and its perturbed version $\tilde{\mathbf{x}}$, the consistency loss of the Π model [157] is defined as the mean squared error (MSE) between the predictions $\hat{y}$ and $\tilde{y}$ corresponding to $\mathbf{x}$ and $\tilde{\mathbf{x}}$. Alongside the supervised loss, which is the cross-entropy (CE) between $\hat{y}$ and the ground truth y when available, the objective function of the Π model is defined as:

$$\mathcal{L} = w \frac{1}{|\mathcal{D}_u|} \sum_{\mathbf{x} \in \mathcal{D}_u} \mathrm{MSE}\left(\hat{y}, \tilde{y}\right) + \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x}, y \in \mathcal{D}_s} \mathrm{CE}(\hat{y}, y). \tag{4.2}$$

Here, $\mathcal{D}_u$ and $\mathcal{D}_s$ represent the subsets of unlabeled and labeled training data, respectively, with the size notation $|\cdot|$ denoting the number of instances in each subset.

Mean teacher The random perturbations observed in the Π model can be inefficient in high dimensions, given that only a limited subset of the input perturbations are capable of pushing the decision boundary into lower density regions [156]. To address this, the mean teacher [152] employs a teacher-student paradigm to enforce consistency during training. The main idea is for the teacher to provide a prediction for the unlabeled input, while the student utilizes this prediction to learn from the consistency loss. To achieve this, two neural networks, namely a student network and a teacher network, are utilized, with $\boldsymbol{\theta}^S$ and $\boldsymbol{\theta}^E$ representing their respective network parameters.

The student network learns in a similar way as the Π model [157], where the difference is the consistency loss, which is defined as the MSE between the predictions of the student and teacher networks for the same input. This encourages the student network to produce consistent predictions with the teacher network. The objective function for the student network is defined as:

$$\mathcal{L} = w \frac{1}{|\mathcal{D}_u|} \sum_{\mathbf{x} \in \mathcal{D}_u} \mathrm{MSE}\left(\hat{y}^S, \hat{y}^E\right) + \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x}, y \in \mathcal{D}_s} \mathrm{CE}\left(\hat{y}^S, y\right). \tag{4.3}$$

In this equation, $\hat{y}^S$ and $\hat{y}^E$ indicate the predictions of the student and teacher networks, respectively.

The teacher network has its parameters updated using an exponential moving average (EMA) of the student network. This update occurs at each epoch, denoted by t, and uses a decay rate, represented by α:

$$\boldsymbol{\theta}_t^E = \alpha \boldsymbol{\theta}_{t-1}^E + (1 - \alpha) \boldsymbol{\theta}_t^S. \tag{4.4}$$

A possible limitation of the mean teacher approach could be the convergence of the teacher model to the student model over many training iterations, which might transfer biased and unstable predictions. An alternative method entails simultaneously training two student models with distinct initializations [158].

Interpolation consistency training Interpolation consistency training (ICT) [151] is proposed to address some of the limitations mentioned above by generating more systematic and meaningful perturbations, which are aligned with the input space. To achieve this, ICT employs a Mixup [159] operator, which interpolates between two instances. The operator creates a perturbed input from two unlabeled inputs as follows:

$$\text{Mix}_\lambda(\mathbf{x}_i, \mathbf{x}_j) = \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j. \tag{4.5}$$

In this equation, $\mathbf{x}_i, \mathbf{x}_j$ are two unlabeled inputs, and $\lambda \sim \text{Beta}(\alpha, \alpha)$ for $\alpha \in [0, \infty]$.

With the perturbed input $\tilde{\mathbf{x}}$, i.e., $\tilde{\mathbf{x}} = \text{Mix}_\lambda(\mathbf{x}_i, \mathbf{x}_j)$, and using the teacher-student paradigm, the objective function for the student network in ICT is defined as follows:

$$\mathcal{L} = w \frac{1}{|\mathcal{D}_u|} \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_u} \text{MSE}\left(\hat{\tilde{y}}^S, \text{Mix}_\lambda\left(\hat{y}_i^E, \hat{y}_j^E\right)\right) + \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x}, y \in \mathcal{D}_s} \text{CE}(\hat{y}^S, y). \tag{4.6}$$

Here, $\hat{\tilde{y}}^S$ represents the prediction of the student network using the perturbed input $\tilde{\mathbf{x}}$, while $\hat{y}_i^E$ and $\hat{y}_j^E$ denote the predictions of the teacher network using inputs $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively.

Notably, the Mixup operator is applied to the teacher network's predictions, i.e., $\text{Mix}_\lambda(\hat{y}_i^E, \hat{y}_j^E)$, where the output is used to calculate the MSE in conjunction with $\hat{\tilde{y}}^S$. The teacher network is a moving averaged version of the student network, as shown in Equation 4.4.

4.1.2 Pseudo-labeling

Pseudo-labeling methods often involve the generation of hard labels for unlabeled data using the prediction function of the network [160]. This enables learning from unlabeled data using the same loss function as used for the labeled data, such as cross entropy [161]. This has an advantage over consistency training using a MSE loss [151, 152, 157], as the cross entropy can produce steeper gradients for better learning from the unlabeled data.

However, the quality of the training largely depends on the accuracy and reliability of the generated pseudo-labels, highlighting the importance of generating high-quality pseudo-labels for optimal training results [153, 162]. [163]

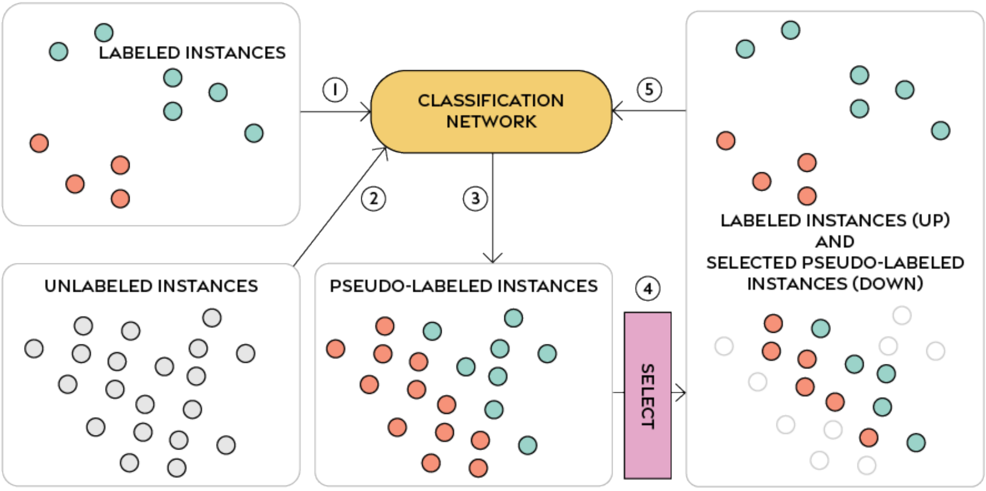


Figure 4.2: An overview of pseudo-labeling.

demonstrates that naive pseudo-labeling can lead to overfitting on incorrect pseudo-labels due to confirmation bias, proposing the use of soft pseudo-labels with regulation techniques instead.

Figure 4.2 provides a step-by-step illustration of the pseudo-labeling process. The network is pretrained on the labeled data, indicated in step ①, and assigns pseudo-labels to unlabeled instances using its predictions, indicated in steps ②-③. A selection mechanism, depicted in step ④, may be employed if needed.

For selection, various strategies can be adopted, such as selecting the top-k unlabeled samples predicted with the highest confidence [164, 165] or using relative confidence based on heuristics [166]. A combination of confidence score and uncertainty can also be utilized to determine which instances to include in training [167–169]. Iscen et al. [170] incorporate label propagation into pseudo-labeling by alternating between training the network on both labeled and pseudo-labeled examples, constructing a nearest neighbor graph, and applying label propagation to improve pseudo-labels. Lastly, the network undergoes further training with the labeled and (selected) pseudo-labeled instances, as shown in step ⑤.

Pham et al. [171] leverage the teacher-student paradigm for pseudo-labeling. The student network learns from the pseudo-labels created by the teacher network. The teacher network, pretrained using labeled data, generates soft pseudo-labels for each unlabeled instance based on its own prediction function. The student network, trained with the pseudo-labels, computes the validation loss to train the teacher network. The gradients of the validation loss in the student network can be backpropagated to the teacher network using policy gradients [172].

4.1.3 Relevance of semi-supervised learning to the included papers

The core idea presented in Papers I and II is based on the simultaneous use of consistency training and pseudo-labeling. In these papers, the predictions for each unlabeled sample are expected to be consistent with those of its neighboring labeled samples.

To enforce consistency, unsupervised clustering [33] is applied to both labeled and unlabeled data, generating a clustering structure with a larger number of clusters than the presented class attributions. This approach allows for a fine-grained investigation of the structure of the data, providing an understanding of the input space without relying on the label space. The pseudo-labels assigned to the entire training data reflect the underlying clustering structure, and the network learns an unsupervised clustering representation based on these pseudo-labels.

Once the unsupervised training phase is complete, the network is trained in a supervised manner on the labeled data to learn class decision boundaries that are aligned with the underlying clustering structure. By alternating between unsupervised and supervised training at the mini-batch level, the network can learn the structure of the input space while taking into account the class decision boundary. This approach allows the network to effectively integrate the unsupervised clustering structure of the data with the labeled class information, leading to improved predictive performance.

4.2 Explainability in deep learning

The application of deep learning in marine environment monitoring has demonstrated impressive progress, as evidenced by recent research works [13, 28, 173–177]. However, a significant limitation of these models is the lack of clarity regarding what information is necessary for the input data and how it should be used to make a decision [178, 179]. If deep learning models are unable to provide an explanation for their decision-making process, their adoption and further application can be limited.

To address this issue, explainability in deep learning has been emphasized, which aims to create intelligent systems that are transparent and interpretable by providing human-understandable explanations for their decisions [26, 27, 41, 180, 181]. This emphasis is expected to lead to the development of new explainable deep learning methods that ensure the safety and reliability of predictions in various scenarios, such as adversarial attacks [182].

Although the importance of explainability is emphasized, the precise definition of explainability and its requirements are still subjects of exploration [26, 41, 95, 178]. The ultimate goal of these requirements is to be mathematically formulated and included in the training process, enabling the network to learn to achieve self-explainability [183]. This differs from *a-posteriori* explainability [184–188] in that their pretrained network often learns with a focus only on input-output relevance, such as cross entropy.

In this section, we will provide a brief comparison between self-explainability and *a-posteriori* explainability, followed by an overview of the requirements for explainability as introduced in the literature [26, 41].

4.2.1 A-posteriori explainability and self-explainability

A-posteriori explainability focuses on interpreting and understanding pretrained neural networks by identifying the input features relevant to the decisions made by a pretrained network [184–188]. The pretrained networks employed in this line of research are frequently optimized to maximize input-output relevance, such as accuracy.

In contrast, self-explainability refers to neural networks where both the network architecture and training mechanisms are specifically designed to offer transparency and interpretability, providing decisions and their reasoning simultaneously [183]. The objective function used in self-explainable models often incorporates explainability requirements that contribute to generating a good explanation. Figure 4.3 presents a visual comparison of *a-posteriori* explainability and self-explainability.

One line of *a-posteriori* methods involves gradient-based approaches, where the explanation is calculated using gradients or reverse propagation in the input space. Examples of this approach include Grad-CAM [187], LRP [184], Guided Backpropagation [185], and DeepLIFT [186]. Grad-CAM [187] uses the gradients of logits, flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image. Although the gradient-based methods are relatively straightforward to implement in the given network architecture, they are limited to the models with differentiable neural activation [189], and the visualizations based on gradient-based methods often contain falsely perceptual regions in addition to a coarse representation [189].

Another line of *a-posteriori* methods is the perturbation-based method, which observes output changes by processing a set of perturbed images [190]. Examples of this approach include LIME [188], RISE [191], Occlusion [192],

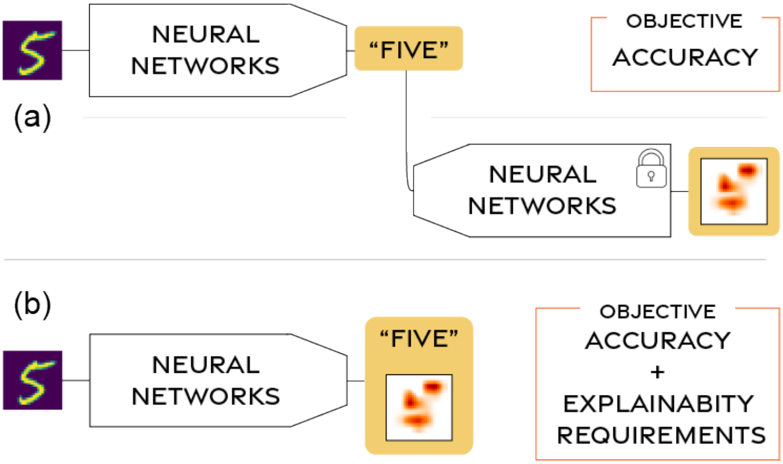


Figure 4.3: Comparison of (a) an *a-posteriori* explainable method and (b) a self-explainable method.

SHAP [193], and real-time detection [194]. A well-known method LIME [188] first employs occlusions of superpixels from the original image to synthesize a number of neighboring image instances. The synthesized instances and the outcomes are used to fit a linear model where the coefficients of the linear model explain the contributions of occluded features. Perturbation-based methods are known for providing robust and reliable explanations in the input space. However, a challenge faced by these methods is the combinatorial complexity explosion, as there may be practical limits in the number of perturbations that can be sampled [190].

Meanwhile, the objective function in self-explainable methods often integrates requirements for explainability, such as intelligibility [41, 195], coherence [183], and minimality [196], into the learning process to account for what constitutes a good explanation. Essentially, this approach leverages the modeling capabilities of neural networks to achieve better explanations [144, 183], transforming black-box neural networks into transparent "glass-boxes" [197] that reveal the decision-making process and the factors contributing to it [198].

While the precise formulation may vary depending on the method and application, one intuitive approach for achieving self-explainability is to mathematically formulate these requirements and add them to the objective function [195, 196], in addition to a loss term seeking input-output relevance. In some cases, modifying the network structure can enhance transparency at specific stages of the decision-making process [183, 196]. For instance, in the self-explaining model SENN [183], the network consists of three modules that explicitly separate the process of extracting relevant features, computing relevance scores, and combining them into a prediction. This modular structure enables the network to achieve self-explainability and increase transparency.

To ensure transparency and interpretability of deep learning models, it is necessary to define and understand the requirements for explainability [26]. One way to categorize these requirements is by using the framework proposed by Sokol et al. [41], which groups them into five categories: usability, functional, operational, safety, and validation requirements. For marine environmental monitoring, it is critical to convey the network's decisions to experts from various disciplines. Therefore, usability requirements are of particular importance in this thesis and will be elaborated on in more detail below.

Sokol et al. [41] propose five groups of requirements for explainability. Functional requirements address algorithmic considerations of the network and explanation, such as supervision level (supervised, semi-supervised, or unsupervised), problem type (classification or regression), explanation target (model predictions, model parameters, or data), and relation to the prediction (*a-posteriori* or self-explainable). Operational requirements indicate the information needed for end-users to effectively interact with explanations, including explanation form (visualization, textualization, or statistical summarization), logical relationship (causal or relevant), and degree of transparency. Safety requirements consider the impact of explainability on the security and privacy aspects of predictive systems, including information leakage and explanation misuse. Validation requirements aim to require a generally agreed-upon validation protocol to assess and prove the effectiveness of the explainability approach. Such a protocol can help eliminate confirmation bias and mitigate selection bias [41, 199].

4.2.2 Usability requirements

Usability requirements adopt a user-centered perspective, focusing on explanation properties that are easily understood by the explainee, as discussed in Sokol et al. [41]. Among several usability requirements that have been identified within the literature, four of these requirements, specifically soundness, parsimony, completeness, and interactiveness, hold particular relevance to this thesis and will be elaborated upon.

Soundness aims to measure the accuracy of an explanation in relation to the underlying prediction model [41, 200]. One potential candidate for measuring soundness is mutual information [36, 201]. This information theoretical metric measures the shared information between two variables, in this case, the mutual information between the explanation and the prediction target.

This metric is referred to as sufficiency within the information bottleneck framework [35], which will be further discussed in Section 4.2.3. By maximizing the mutual information, the network learns to increase the relevance of the information contained in the explanation concerning the target, leading to more accurate predictions.

The parsimony requirement emphasizes that explanations should be succinct and focused, avoiding the unnecessary inclusion of information that may overwhelm the explainee [26, 41, 200, 202]. Parsimony can be employed as a strategy to decrease the complexity of the explanation, ensuring that the model's description is suitable for users with varying levels of background knowledge.

Within the information bottleneck framework [35], this requirement is referred to as minimality, and is mathematically formulated to minimize the mutual information between the input and the explanation [35, 36, 201]. By integrating the soundness and parsimony requirements, a good explanation can be defined as one that effectively conveys the most information while using the fewest arguments [196].

Completeness pertains to the degree to which an explanation can generalize and accurately represent the underlying predictive model [26, 41]. A dependable and effective explanation should have the ability to generalize well beyond a specific sample, indicating that the explainability method should provide reliable explanations for all samples within the dataset. This requirement is closely related to self-explainability, which strives to attain a global explanation through learning.

Interactiveness is a vital component of explanation fidelity, as user experiences can greatly differ based on their domain knowledge and expertise level [203]. To enhance the user experience, the explanation process should be controllable and interactive, enabling users to tailor the explanation according to their individual needs [26, 41, 202, 204]. In the context of multi-frequency echosounder data, which is inherently multi-disciplinary, the network's explanation becomes more persuasive when it incorporates knowledge from other relevant disciplines.

4.2.3 Information bottleneck

Recent advancements in explainable deep learning methods [149, 205–207] have been leveraging information theory [37, 38, 208] to enhance the interpretability and transparency of deep learning models. These methods utilize metrics of information quantities and information theoretical learning principles to explain the underlying mechanism of the model's decision.

The information bottleneck (IB) framework [36, 149, 207] has emerged as a promising approach in this context. It aims to find the optimal bottleneck representation T between the input X and the output Y [35, 36, 207]. This is accomplished by balancing two opposing objectives: minimizing the mutual information between X and T for the explanation's minimality, e.g., $I(X;T)$, and maximizing the mutual information between T and Y for sufficiency. The first goal ensures that the bottleneck representation T contains as sufficient information of the input X as possible. Meanwhile, the second goal aims to compress the bottleneck T to have minimal but essential information relevant to the output Y. The general objective function of the IB framework, denoted by $\mathcal{L}$, is defined as:

$$\mathcal{L} = I(T;Y) - \beta I(X;T), \tag{4.7}$$

where this trade-off between sufficiency and minimality can be controlled by a Lagrange multiplier β.

Information bottleneck explainability The IB framework has recently gained attention as a promising approach for achieving explainable deep learning [196, 209]. This is due to the sufficiency and minimality components in the IB framework, which align with the requirements for explainability outlined in [41, 182]. Furthermore, by optimizing the objective function in Equation 4.7 using gradient descent, the network can become self-explanatory [183, 196].

To enable explainability, an attribution mask denoted as M is introduced. Each pixel in M represents the importance score of the corresponding pixel in the input X in making a prediction. The attribution mask M is integrated with the IB framework through the bottleneck representation T:

$$T = M \odot X, \tag{4.8}$$

where $\odot$ denotes the Hadamard product that performs element-wise multiplication.

By combining Equation 4.8 with Equation 4.7, the result is:

$$\mathcal{L} = I(M \odot X;Y) - \beta I(X;M \odot X). \tag{4.9}$$

The presented learning procedure involves two network modules, the explainer and classifier, connected in series to realize the objective function in Equation 4.9. During the forward pass, the explainer module creates the attribution mask M from the input X, where each pixel is restricted to have a score between 0 and 1. Using the Hadamard product in Equation 4.8, the mask representation M simulates spatial feature removal by partially or completely masking elements of X. The subsequent classifier module then classifies the bottleneck representation T into classes in the label Y. During the backward pass, both network modules are simultaneously optimized in an end-to-end fashion using gradient descent. Figure 4.4 illustrates an overview of the IB explainability.

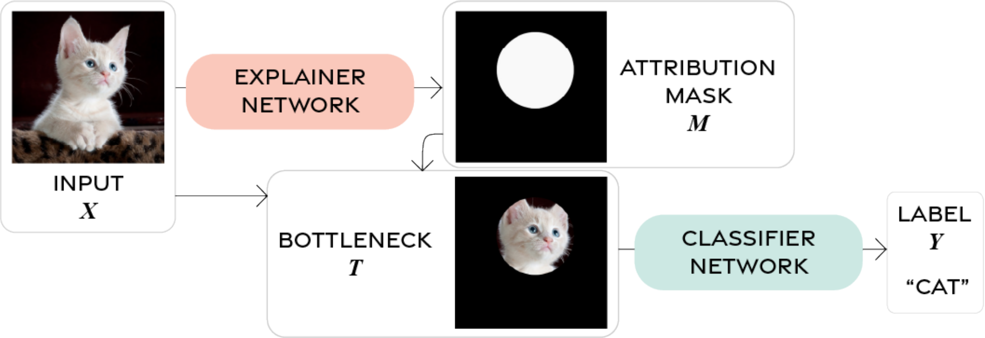


Figure 4.4: Visual illustration of the IB explainability on an image.

4.2.4 Relevance of explainability to the included paper

Paper III proposes a novel explainability method, DIB-X, which stands for deterministic information bottleneck explainability. DIB-X is a generic self-explainable deep learning method that addresses the usability requirement of explanation. It is inspired by the multi-disciplinary nature of marine environmental monitoring and the necessity to ensure reliability for experts from diverse background domains.

DIB-X leverages the IB framework [35] to address the usability requirements of soundness and conciseness, which correspond to sufficiency and minimality, respectively, within the framework. Additionally, DIB-X seeks to improve interactivity by incorporating domain knowledge into the explanation. The completeness requirement is also addressed as DIB-X operates in a self-explainable manner.

In addition, DIB-X proposes a novel solution that directly computes mutual information without relying on variational methods to address the challenge of computing mutual information in a high-dimensional space [35]. The mutual information between the input and the explanation, denoted as $I(X;T)$, is widely acknowledged as a difficult task due to its intractable nature [196, 207, 209]. Existing IB approaches [210–212] often rely on variational approximation or adversarial training to maximize a lower bound of the original IB objective. However, Chen et al. [209] argue that these lower bounds may not be tight in practice, especially when training data is limited. Therefore, by directly computing mutual information, DIB-X offers an alternative approach for the IB-based explainable method that does not rely on variational approximation.

Part II

Summary of Research and Concluding Remarks

5 | Paper I

Semi-supervised Target Classification in Multi-frequency Echosounder Data

Changkyu Choi, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, Olav Brautaset, Line Eikvil, and Robert Jenssen

ICES Journal of Marine Science, vol. 78, no. 7, Oct. 2021

Summary

The paper aims to address a limitation of supervised learning, which is the heavy dependence on manually annotated data instances. For situations with limited access to annotated data, a novel semi-supervised deep learning method is proposed, which employs a small number of annotated instances along with a large amount of unannotated instances within a single end-to-end trainable convolutional neural network architecture. The proposed method functions with two interconnected objective functions, e.g., a clustering objective and a classification objective. These objectives optimize the shared convolutional neural network alternately. The clustering objective leverages the underlying structure of all data, both annotated and unannotated, while the classification objective enforces consistency within given classes using the limited annotated instances.

Figure 5.1 provides an overview of the proposed method. In the figure, each point represents an extracted patch, with gray points being unannotated and colored points (red, green, or blue) indicating annotated patches corresponding to their class. (a) The training data occupies an arbitrary space. (b) The clustering objective helps form clusters regardless of annotation. (c) The available annotated data and the classification objective optimize the CNN in a supervised manner. (d) By iterating through steps (b) and (c), the method constructs a decision boundary with respect to given classes, where unan-

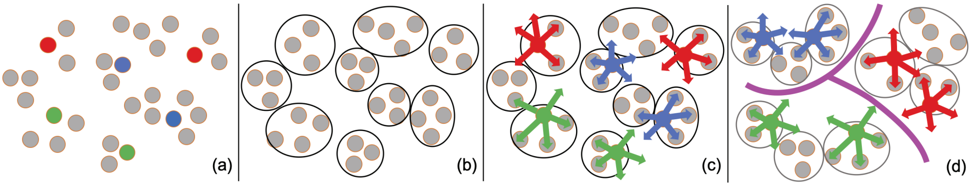


Figure 5.1: Overview of the proposed method in Paper I.

notated points are positioned within the boundary according to their respective clusters.

Our proposed method is methodologically versatile and has been evaluated in the context of acoustic target classification (ATC), a field of significant interest for marine ecosystem and fishery management due to its potential to estimate species abundance or biomass. We assess the method using multi-frequency echosounder data from a sandeel case study in the North Sea. The experimental results demonstrate the effectiveness of our method in this application.

Contributions by the author

In this collaborative research project, the main idea of the proposed method was jointly conceived by myself, Michael Kampffmeyer, Arnt-Børre Salberg, and Robert Jenssen. As the first author, I was responsible for preprocessing the echosounder data, implementing the proposed method, conducting experiments, writing the manuscript, and finalizing the paper. At each stage of the project, I actively sought and integrated feedback and revisions from my co-authors, which contributed to the rigor and quality of the final paper.

6 | Paper II

Deep Semi-supervised Semantic Segmentation in Multi-frequency Echosounder Data

Changkyu Choi, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, and Robert Jenssen

IEEE Journal of Oceanic Engineering, vol. 48, no. 2, 2023

Summary

This paper proposes a novel semi-supervised semantic segmentation method, which is an extension of the semi-supervised deep learning method proposed in Paper I, tailored for the semantic segmentation task. The fisheries and aquatic industry has a specific interest in semantic segmentation, as it enables non-invasive estimation of marine organism abundance and large-scale observation of the underwater environment. However, the high degree of class imbalance in semantic segmentation, where the background class accounts for approximately 99 percent of total pixels, presents a considerable challenge. To tackle this issue, the proposed method incorporates a class-balancing technique based on the model's predictions into the learning process, alongside the alternating optimization proposed in Paper I. The proposed semi-supervised segmentation method achieves results comparable to the standard supervised semantic segmentation method while utilizing a smaller amount of annotated data.

Figure 6.1 provides an overview of the proposed method. In the figure, (a) Input echosounder data, where each backscattering intensity (pixel) will be clustered and classified into given classes. (b) Clustering structure identified by the unsupervised clustering objective. This clustering structure serves as the pseudo-label to train the network using cross-entropy. (c) Pixel-level ground-truth annotation. The supervised segmentation objective leverages this to optimize the network in a supervised manner. In a semi-supervised setting, only a few input images have these annotations. (d) Predicted segmentation

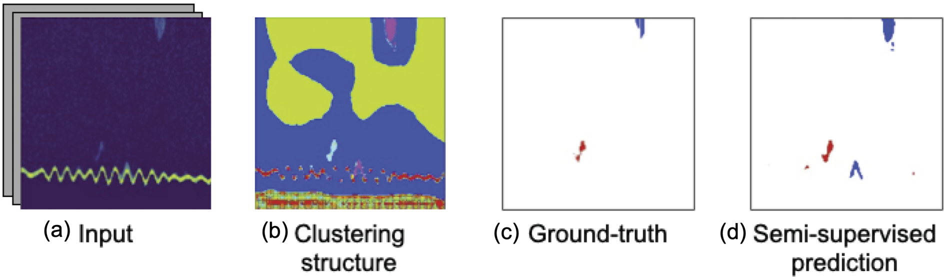


Figure 6.1: Overview of the proposed segmentation method in Paper II.

map achieved by alternating between the two objectives.

Contributions by the author

In this collaborative research project, the main idea of the proposed method was jointly conceived by myself, Michael Kampffmeyer, Arnt-Børre Salberg, and Robert Jenssen. As the first author, I was responsible for preprocessing the echosounder data, implementing the proposed method, conducting experiments, writing the manuscript, and finalizing the paper. At each stage of the project, I actively sought and integrated feedback and revisions from my co-authors, which contributed to the rigor and quality of the final paper.

7 | Paper III

Deep Deterministic Information-Bottleneck Explainability on Marine Image Data

Changkyu Choi, Shujian Yu, Michael Kampffmeyer, Nils Olav Handegard,
Arnt-Børre Salberg, and Robert Jenssen
Submitted to *Pattern Recognition*

Summary

This paper introduces DIB-X, a novel self-explainable method that emphasizes usability principles of explanations [41], represented as minimality, sufficiency, and interactivity. These principles are mathematically formulated in the objective function, allowing the network to learn the principles during optimization. The minimality and sufficiency principles are formulated through the information bottleneck (IB) framework, which seeks to find the optimal balance between their trade-off relationship. Notably, DIB-X directly quantifies the minimality principle using the recently proposed matrix-based R'enyi's α-order entropy functional, eliminating the need for variational approximation. The interactivity principle is achieved by incorporating existing domain knowledge as prior explanations, promoting explanations that align with established domain understanding. Empirical results on two marine environment monitoring datasets demonstrate the effectiveness of the proposed method.

Figure 7.1 offers an overview of the proposed method, where the method consists of four steps. ① The input image X is used to create the attribution mask M by being processed by the *explainer* network module. ② If domain knowledge in the form of the mask prior M_p is available, it can be integrated into the attribution mask. ③ The attribution mask is employed to generate the bottleneck representation T, which is obtained by taking the Hadamard product of the mask with the input image X. ④ The *classifier* network module processes the bottleneck representation T to perform the classification task.

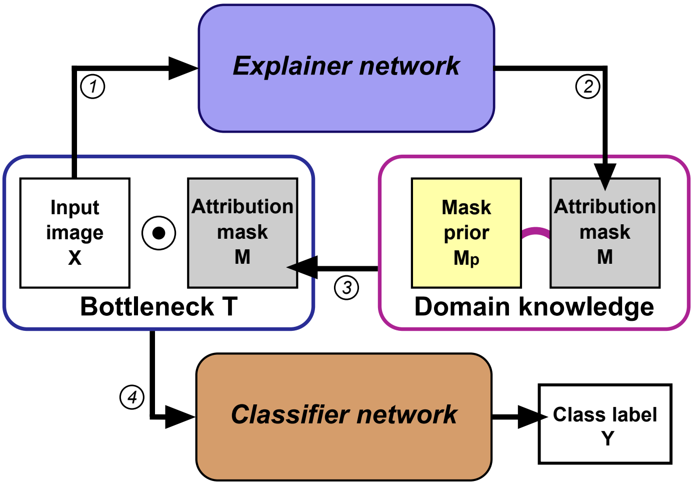


Figure 7.1: Overview of the proposed DIB-X in Paper III.

Contributions by the author

In this collaborative research project, the main idea of the proposed method was jointly conceived by myself, Shujian Yu, Michael Kampffmeyer, Arnt-Børre Salberg, and Robert Jenssen. As the first author, I was responsible for preprocessing the echosounder data, implementing the proposed method, conducting experiments, writing the manuscript, and finalizing the paper. At each stage of the project, I actively sought and integrated feedback and revisions from my co-authors, which contributed to the rigor and quality of the final paper.

8 | Concluding Remarks

This thesis aims to advance deep learning for marine environmental monitoring by addressing key challenges such as limited annotated data and explainability, which hinder its broader application in the field.

To address the challenge of limited annotated data, we propose a generic semi-supervised classification method that can be extended to semi-supervised semantic segmentation. Our approach is assessed using multi-frequency echosounder data, demonstrating performance on par with fully-supervised methods while requiring fewer annotated instances. This data-efficient method notably reduces annotation costs and maintains performance, showcasing its potential for real-world applicability.

To address explainability, we propose a self-explainable deep learning method, DIB-X, which simultaneously provides decisions and explanations throughout the learning process. Inspired by the information bottleneck framework [35], DIB-X identifies latent bottleneck representations as explanations while balancing information sufficiency and minimality. Additionally, DIB-X can integrate domain knowledge as prior information, facilitating explanation learning based on existing knowledge.

Enhanced explainability from DIB-X fosters trust among domain experts, which is crucial in the multidisciplinary field of marine environmental monitoring. By improving model explainability, we facilitate the integration of deep learning models into marine monitoring practices, potentially paving the way for more transparent and reliable solutions for marine environment preservation.

Limitations and future work

While the thesis has made contributions to addressing key challenges in deep learning for marine environmental monitoring, some limitations and potential future work still remain.

Paper I Our semi-supervised learning method is designed to train the network by alternately optimizing two objective functions. However, this alternation may lead to confusion in the initial stages of training, resulting in inefficiency and slow convergence.

To address this challenge, future work could explore simplifying the training process by merging both objectives into a single function. This approach would allow the network to simultaneously focus on both classification and clustering tasks. A promising method to achieve this unification involves adopting the mathematical formulation introduced by Boubekki et al. [213], which enables unsupervised clustering through gradient descent.

By implementing this method, we can create a unified objective function for semi-supervised learning, which in turn enhances the training process of our proposed method. Moreover, this can contribute to the generalization of semi-supervised learning, making them applicable to a wide variety of problems.

Paper II Our semi-supervised segmentation method addresses the challenge of severe class imbalance by utilizing weighted cross entropy, where the weight is internally determined by the model's predictions. While this approach attains a satisfactory level of predictive performance, there remains potential for further investigation and improvement.

Future work could investigate alternative loss functions known for their robustness in handling class imbalance, in order to assess their impact on performance. Additionally, we could explore incorporating new metrics based on information theory [37], such as mutual information [36], into the objective function.

From a practical perspective, a follow-up study comparing various loss functions to select the most appropriate one would be a valuable contribution, as this has not yet been well-established in the field of marine environment monitoring, particularly in acoustic target classification.

Paper III In this work, we have demonstrated that DIB-X performs well on uni-modal data from different collection modalities, such as RGB cameras and echosounders. As a result, it is a natural progression to extend its application to multi-modal data in future research.

In practice, while collecting echosounder data, biological sampling might be conducted by deploying a trawl in areas with strong backscattered intensity, or by capturing images with an RGB camera attached to the trawl. Consequently, it would be valuable to utilize this data to investigate multi-modal explainability.

Concluding remark

In conclusion, this thesis highlights the imperative for collaboration between deep learning experts and domain specialists in the realm of marine environmental monitoring, emphasizing the necessity of interdisciplinary approaches to tackle intricate real-world challenges. The methods proposed herein represent the commencement of this endeavor, with subsequent steps involving evaluation, integration, and operation being crucial to their effective implementation in the field. It is anticipated that as deep learning and marine environment monitoring further develop and converge, this thesis will serve as a foundational work, stimulating additional progress in this critical area of research. Through the cultivation of robust collaborations and the adoption of pioneering methodologies, we can collectively contribute to the preservation and protection of our invaluable marine ecosystems.

Part III

Included Papers

9 | Paper I

ICES Journal of
Marine Science

ICES CIEM
International Council for the Exploration of the Sea
Conseil International pour l'Exploration de la Mer

ICES Journal of Marine Science (2021), 78(7), 2615–2627. https://doi.org/10.1093/icesjms/fsab140

Original Article

Semi-supervised target classification in multi-frequency echosounder data

Changkyu Choi [1], Michael Kampffmeyer[1,2], Nils Olav Handegard [3], Arnt-Børre Salberg[2], Olav Brautaset [2], Line Eikvil,[2], and Robert Jenssen[1,2,*]

[1]*UiT The Arctic University of Norway, P.O. Box 6050, Langnes Tromsø 9037, Norway*
[2]*Norwegian Computing Center, P.O. Box 114, Blindern, Oslo 0314, Norway*
[3]*Institute of Marine Research, Nordnesgaten 50, Bergen 5005, Norway*

[*]*Corresponding author: tel: (+47) 77646493; e-mail: robert.jenssen@uit.no*

Choi, C., Kampffmeyer, M., Handegard, N. O., Salberg, A.-B., Brautaset, O., Eikvil, L., and Jenssen, R. Semi-supervised target classification in multi-frequency echosounder data . – ICES Journal of Marine Science, 78: 2615–2627.

Received 8 November 2020; revised 20 May 2021; accepted 8 June 2021; advance access publication 12 August 2021.

Acoustic target classification in multi-frequency echosounder data is a major interest for the marine ecosystem and fishery management since it can potentially estimate the abundance or biomass of the species. A key problem of current methods is the heavy dependence on the manual categorization of data samples. As a solution, we propose a novel semi-supervised deep learning method leveraging a few annotated data samples together with vast amounts of unannotated data samples, all in a single model. Specifically, two inter-connected objectives, namely, a clustering objective and a classification objective, optimize one shared convolutional neural network in an alternating manner. The clustering objective exploits the underlying structure of all data, both annotated and unannotated; the classification objective enforces a certain consistency to given classes using the few annotated data samples. We evaluate our classification method using echosounder data from the sandeel case study in the North Sea. In the semi-supervised setting with only a tenth of the training data annotated, our method achieves 67.6% accuracy, outperforming a conventional semi-supervised method by 7.0 percentage points. When applying the proposed method in a fully supervised setup, we achieve 74.7% accuracy, surpassing the standard supervised deep learning method by 4.7 percentage points.

Keywords: acoustic target classification, deep clustering, limited annotation, pseudo-labeling, semi-supervised deep learning

Introduction

Acoustic target classification is a field of research that analyzes the marine acoustic data for the marine ecosystem and fishery management, and an analysis task of multi-frequency echosounder data is a major interest (Korneliussen, 2018). The goal is to assign an observed acoustic backscattering intensity to a given acoustic category. The results can be used to estimate the abundance or biomass of the species (MacLennan and Simmonds, 2013).

One common approach for acoustic target classification is manual categorization, where the operators identify and select regions with similar acoustic properties (Korneliussen, 2018). This manual categorization may be supported by relative frequency response (Kloser *et al.*, 2002; Korneliussen and Ona, 2003), echo traces (Reid, 2000), trawl sampling (Handegard and Tjøstheim, 2009), and domain knowledge of the target categories. However, the application of the supporting methods is limited due to their extremely high cost, making the manual process vulnerable to bias from the operators. Hence, automated and scalable analysis methods are required to efficiently cope with the multi-frequency data.

Deep learning, a family of data-driven computational models known for their flexibility and scalability, can provide an answer to the need. Especially convolutional neural networks (CNNs), a popular deep learning framework, are renowned to excel at image tasks (Long *et al.*, 2015). Although echosounder data are not images in the traditional sense, there exist commonalities between the two. Both data sources reflect visual observations, where each observation channel provides a structured form of the data in a two-dimensional array. Based on the commonality, a few studies have successfully applied the CNNs to perform target classification on the echosounder data, where the tasks are detection of sandeel (SE) schools (Brautaset *et al.*, 2020) and herring schools (Rezvanifar *et al.*, 2019). These CNNs learn how to extract abstract characteristics from patterns in the echosounder data, and the extracted characteristics are referred to as feature representation.

The feature representation that the neural networks learn is dependent on the formulated objective function. The objective function is designed to reflect the goal of the task, and measures an error between the current prediction of the CNN and the optimum that is often the human-provided annotation. "Fully supervised learning" refers to algorithms where the entire training data set is annotated. The learning scheme of the CNN is an iterative optimization process that gradually minimizes the error measured by the objective function. Provided a high quality of the training data and that an appropriate choice of the CNN are assured, the fully supervised learning approaches achieve a good level of performance as the model learns the feature representations in a way to mimic the corresponding annotations of the data.

It is, however, extremely costly and challenging to acquire the annotations in many real-world data including the echosounder data. The aforementioned acoustic target classification studies using CNNs learn in a fully supervised fashion, which heavily depends on the manual categorization process by the operators in order to train their models. Hence, new learning schemes are required in order to deal with an increasing volume of the datasets in an efficient and effective manner, where the dependency on the annotated data is reduced.

In this paper, we propose a novel deep learning algorithm for acoustic target classification, which operates on the condition that only a small part of the data is annotated, referred to as semi-supervised deep learning (Chapelle *et al.*, 2009). The novelty of our work is that the proposed algorithm exploits the underlying structure of the data including both the annotated part and the unannotated part using two interconnected objective functions, namely, a clustering objective and a classification objective. The alternating optimization process by the two objective functions allows the unannotated part of data to contribute to form decision boundaries with respect to the given classes, which is not applicable for a common supervised deep learning (SDL). To the best of our knowledge, this is the first semi-SDL algorithm applied for the acoustic target classification.

The multi-frequency echosounder data used in this study have been annually collected at the North Sea since 2009 by the Norwegian Institute of Marine Research for the case study of classifying lesser SE (*Ammodytes marinus*), a small fish without a swim bladder. Due to the abundance and fat richness (Raitt, 1934), it is considered as the major forage fish of the food chain, preyed on by a great variety of predators such as piscivorous fish species, marine mammals, and seabirds (Daan *et al.*, 1990; Furness, 2002). Analogously, the depletion of the SE stock causes a severe damage to the ecosystems (Johnsen *et al.*, 2017). For instance, Frederiksen *et al.* (2007) argue that there were exceptionally high breeding failures for most seabird species in the North Sea in 2004, due to a sharp decline of SE stocks in 2003, where the annual landing of SEs in 2003 was reduced to approximately 40% of the average landings in the ten previous years (ICES, 2017). The proposed method considerably reduces the dependency on the annotated data and contributes to the automated SE stock estimation, which is important for the ecosystems as well as the fisheries in the North Sea.

Extensive experiments conducted on this SE echosounder data validate the robustness of the proposed method. Regarding the patch-level semantic segmentation task, which classifies small and fixed-shaped patches extracted in a regular grid from the multi-frequency echosounder data, the proposed method outperforms both the semi-supervised benchmark under the partially annotated condition and the standard SDL under the fully annotated condition.

The contributions of this article are (i) to develop a novel semi-SDL algorithm that is suitable for segmenting and classifying echosounder data without prior information, and (ii) to demonstrate the proposed algorithm on a real test case.

Background and material

Echosounder data collection

In every April and May since 2005, The Norwegian Institute of Marine Research has conducted acoustic trawl surveys in the SE areas of the North Sea (Johnsen *et al.*, 2017). The SE echosounder data are measured during the surveys by multifrequency Simrad EK60 echosounder systems operating at four different frequency channels (18, 38, 120, and 200 kHz) on the vessel whose speed was approximately 10 knots. The echosounders were calibrated in accordance with the standard procedures before each survey. See Johnsen *et al.* (2009) for further details.

For each frequency channel, a volume backscattering coefficient s_v, an average amount of backscattering intensity per cubic metre (MacLennan *et al.*, 2002), is stored as a corresponding pixel value of the two-dimensional echosounder data. The data are collected at 1 Hz. The horizontal length of a single pixel is 1 second and the vertical length of a single pixel is 19.2 centimeters based on the pulse duration of 1.024 milliseconds. The height and width of the echosounder data, therefore, depends on the depth of the sea and the navigating time for the survey. We analyze echosounder data that have been collected between 2011 and 2019. The average height of the echosounder data is 399 pixels, corresponding to 76.6 meter depth. The total navigation time is 2,407 hours, which is approximately 11 days per year. For cross-validation, we split the data into two groups by year and assign the data between 2011 and 2017 to the training set, and the data from 2018 to 2019 to the test set.

Preprocessing and pixel-level annotation

In the preprocessing phase, all the volume backscattering values s_v are transformed in a decibel unit (dB re 1m^{-1}). The values less than -75 dB re 1m^{-1} or greater than 0 dB re 1m^{-1} are set to -75 dB re 1m^{-1} or 0 dB re 1m^{-1}, respectively. Infrequently, a few number of columns of the data are missing due to the temporary poor reception of the echosounder. We impute the minimum value -75 dB re 1m^{-1} to the missing columns with respect to a common time-range grid based on the resolution of the 200 kHz echosounder data. Pixels with NaN (not a number) are also replaced with -75 dB re 1m^{-1}. We leverage both pixel-level annotation and preprocessing methods from the earlier work (Brautaset *et al.*, 2020), for which we share the echosounder data.

Each pixel in the echosounder data is annotated into three classes based on the frequency response, where the classes are SE, other fish species (OT), and background (BG). An expert operator manu-

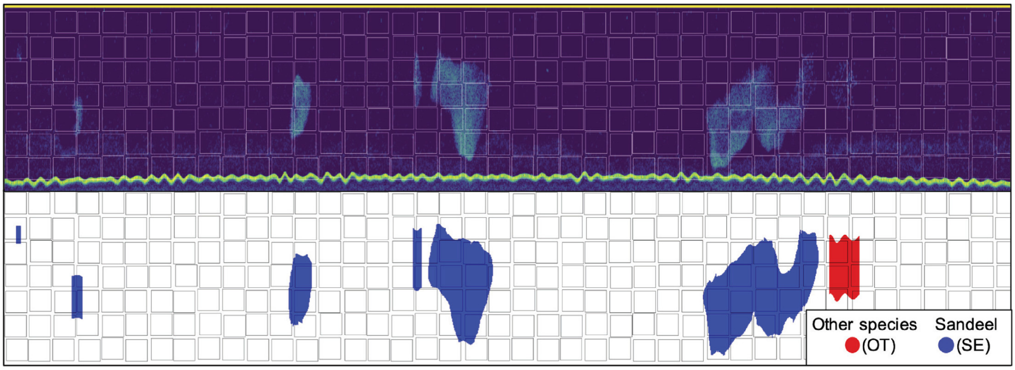


Figure 1. A part of the echosounder data at 200kHz (**up**) and the corresponding pixel-level annotation map (**down**). Each square indicates a patch of size 32 × 32 pixels, where the squares are regularly overlaid on the echosounder data with a random shift in a range of [−2, 2] pixels both horizontal and vertical axes. Each echosounder patch and the corresponding annotation patch are extracted from the same location. Patches having a surface effect (yellow line at the top of the echosounder data) are discarded. SE are colored blue and the school of OT is red in the pixel-level annotation map.

ally delineates the fish school boundaries and annotates the schools across all years using the Large Scale Survey System software (Korneliussen *et al.*, 2016). The primary frequency for the software is chosen to 200 kHz considering the highest SE signal-to-noise ratio (Johnsen *et al.*, 2009). The operator adjusts the detection threshold centered at −63 dB at the primary frequency to visually distinguish the fish school boundaries. The delineated boundary is refined using binary morphological closing to have smoother and realistic edges (Brautaset *et al.*, 2020). The species decision process of the delineated fish schools is also manually performed by inspecting the frequency response for each detected school and is further validated by trawl samples where applicable. In addition to the expensive manual process, there is an element of tacit knowledge as with any expert system. This challenges to reliably define the criteria for the classification, as an effect from the operator may implicitly influence the decision.

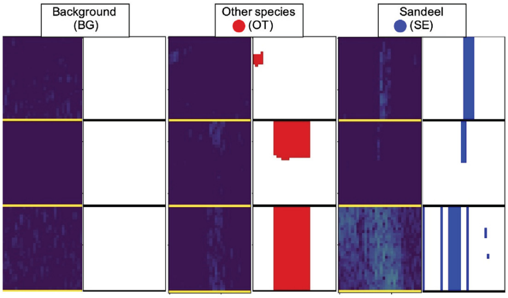


Figure 2. Nine pairs of the patches extracted from the echosounder data at 200 kHz and the corresponding pixel-level annotation map. Three patches are randomly selected per class of BG, OT, or SE.

Patch extraction and annotation

In general, CNN-based image tasks assume a fixed dimension of both an input image and the outcome. To apply CNN on the echosounder data, we extract fixed and small-sized patches from the data. Each extracted patch consists of 32 × 32 × 4 pixels, where "4" refers to the number of echosounder channels. This patch classification task can be seen as a down-stream task since the CNN learns visual features from the patches, and abstracts the learned features to class prediction vectors, where the length of the vector is equal to the number of classes to predict. Note that each element in the vector represents the probability of the class prediction of the patch with respect to each class that is achieved by the softmax function (see deep learning terminologies in the Appendix for the further details).

For the training patch extraction, we administer two criteria to avoid potential sources of bias: overlap between patches is not allowed, and the extracting location of a patch should be determined with stochasticity. Abiding by the criteria, we first overlay grid points spacing 36 × 36 on both the echosounder data and the corresponding pixel-level annotation map. Figure 1 depicts the overlay of the windows for patch extraction based on the grid points. Each grid point becomes the center of the window for the patch extraction, randomly shifted within a range of [−2, 2] pixels to both width and height axes to add stochasticity. Due to the margin in the spacing of the overlaid grid points, there is no overlap between patches. Note that the stochastic spacing is only applied to the training set. The patches from the test set are extracted from a fixed grid, where the centroids are spaced in 32 × 32. To neglect the undesired surface effect that lies at the first ten rows from the top of each echosounder data, we locate the grid points in a way that patches exclude this surface effect. Figure 2 shows the patches from the echosounder data and the pixel-level annotation map.

We annotate each echosounder patch leveraging the corresponding pixel-level annotation. According to the extracted patch dimension, 1024 (32 × 32) annotated pixels determine the patch annotation. We assign the SE or OT class to the patch, where the number of corresponding fish pixels is greater than or equal to 16 pixels which occupy 1.56% of the pixels in the patch. On the other hand, the patch without fish-annotated pixels is annotated to the BG class. The number of patches having both SE and OT pixels together or one fish class but less than 16 fish pixels is negligibly small and those patches are discarded.

Table 1. Extracted patches from the training echosounder data (2011–2017), and the test echosounder data (2018–2019).

Year	**Training set (2011–2017)**		**Test set (2018–2019)**	
Class	**Extracted patches (percentage)**	**Undersampled patches**	**Extracted patches (percentage)**	**Undersampled patches**
BG	1 200 075 (97.81)	10 922	816 726 (97.61)	6 004
OT	15 965 (1.30)		6 004 (0.72)	
SE	10 922 (0.89)		13 984 (1.67)	
Total	1 226 962 (100.00)	32 766	836 714 (100.00)	18 012

Table 1 represents the number of patches extracted from the echosounder data. Severe class imbalance is observed, with more than 97% of the patches belonging to the BG class. To tackle the class imbalance, we randomly undersample patches from the majority classes to obtain the same number of patches for each of the classes (Buda *et al.*, 2018), resulting in a total number of training patches of 32766, and a total number of test patches of 18012. The patches that are excluded from both the training set and the test set are leveraged for tuning hyperparameters.

Deep clustering

We present a novel semi-SDL method, where the idea of the proposed method is to exploit *both* the intrinsic structure of the data and the available annotation, in a single CNN. This method can be applied to the echosounder data as well as being potentially generalized to other data sources since it incorporates the generic idea of deep clustering into the SDL.

Deep clustering refers to unsupervised deep learning based approaches, that aim to cluster data into underlying groups without requiring the class attributes of the data (Korneliussen, 2018). It leverages the representation power of the neural network in conjunction with clustering algorithms, and partitions the input data into clusters with respect to the learned representation. As clustering performance heavily depends on the underlying structure of the data, deep clustering leverages the neural network to encode the training images in the feature representations where the clustering task becomes much easier (Jabi *et al.*, 2019).

There are two main directions of deep clustering with respect to designing the objective function, namely, cluster-discriminative and cluster-generative objectives. Using mutual information or divergence measures, models with cluster-discriminative objectives learn the decision boundaries in-between clusters via posteriors over the assignments given the inputs (Jabi *et al.*, 2019). Deep divergence-based clustering (DDC) exemplifies this line of research (Kampffmeyer *et al.*, 2019), where the objective of DDC is designed to increase divergence between clusters while achieving compactness within a cluster using information-theoretic divergence measures. Deep clustering models that utilize cluster-generative objectives, such as *k*-means, have also been studied (Caron *et al.*, 2018; Biernacki *et al.*, 2000). In their model, referred to as DeepCluster, they explicitly model the density of datapoints within the clusters via likelihood functions. For a given image dataset, the *k*-means clustering models *K* different densities, where each density refers to an image descriptor or a visual feature. This has the advantage that it is easy to increase the capacity of more visual features by simply increasing the number of clusters *K*, leading to all-purpose visual features.

The scalability of the visual features in the DeepCluster is the reason why our method takes its main inspiration from Caron *et al.* (2018) when analyzing the echosounder data. The echosounder patches have many sources that can cause a large variance within their feature representations. Examples include the type of fish, the arrangement and density of the fish pattern, and the location and the occupied area of the fish pattern inside the patch, to name a few. The method of Caron *et al.* (2018) enables to partition the feature representations across the numerous sources of the variance into many clusters, and eventually discovers the intrinsic structure of the data.

However, there is potentially valuable information given by even just having a few annotations and it is crucial to be able to leverage this information. Hence, we propose a new approach that has the capability to also exploit annotated data, even in small amounts.

Method

Objective functions

The key novelty of this paper is to propose a new type of deep neural network leveraging vast amounts of unannotated data (unsupervised) while being able to simultaneously exploit some available annotated data (supervised), yielding a novel *semi-SDL* algorithm. This is achieved through the optimization of an unsupervised clustering objective in addition to a supervised classification objective as outlined in Figure 3. The alternating optimization process enables a CNN that is trained through two interconnected objective functions.

The clustering objective, which utilizes ideas from the study of Caron *et al.* (2018), exploits the underlying structure of the data using *k*-means without requiring any annotation. *The classification objective* enforces consistency of predictions with regards to the given classes in the annotated data. These objectives optimize the CNN in an alternating manner. Through our alternating optimization procedure, we further indirectly incorporate the annotation information into the model, influencing the clustering objective to learn both a structured representation as well as a representation that is consistent with the available annotations. Figure 4 outlines the learning procedure that is further described below.

Clustering objective

Refer to the Appendix for detailed information of the terminologies, such as a cross-entropy loss, end-to-end learning, softmax, and epoch. The clustering objective of our proposed semi-supervised model aims to address both the clustering of the input data as well as the optimization of the CNN.

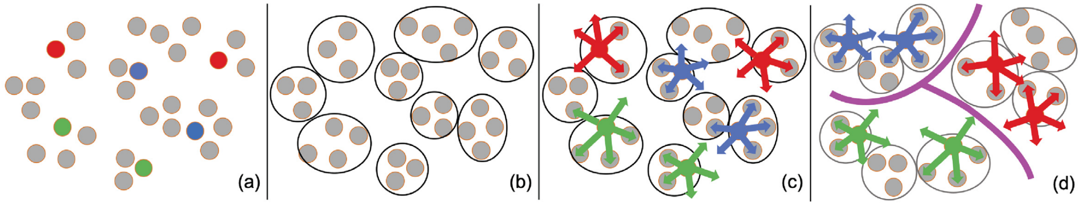


Figure 3. Overview of the proposed method. Each point represents the extracted patch, where the point in gray is unannotated while the points in color (red, green, or blue) indicates the annotated one with respect to the class. (a) The training data occupy an arbitrary space. (b) The clustering objective helps to form clusters regardless of the annotation. (c) The available annotated data and the classification objective optimize the CNN in a supervised manner. (d) The iteration of (b) and (c) constructs the decision boundary with respect to given classes, where the unannotated points take their place inside the boundary according to their own clusters.

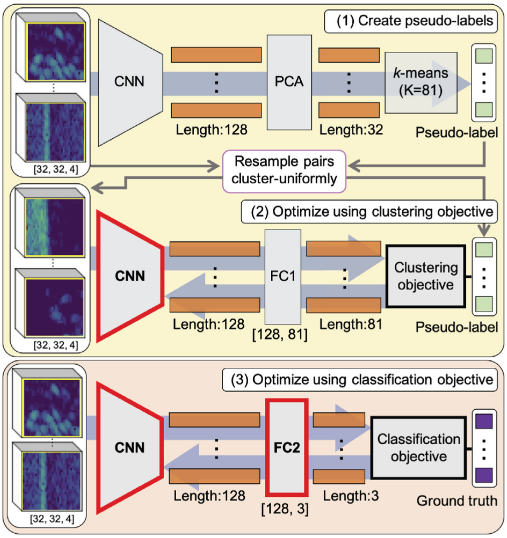


Figure 4. Training procedure of the proposed method. Each orange bar represents the feature representation of each patch in a vectorial form of the specified length. We configure that the output of the CNN is a vector of length 128. Only the CNN and FC2 layer with red outlines are optimized in each stage. (1) Create pseudo-labels. (2) Optimize the CNN using cluster objective. (3) Optimize the CNN and FC2 using classification objective.

The proposed method takes inspiration from the study of Caron *et al.* (2018) that clusters using k-means and optimizes the CNN based on the cluster assignments, which are called pseudo-labels. The proposed method clusters the feature representations of all training patches into K clusters using k-means, in a way to find the best assignment that minimizes the k-means loss:

$$\mathcal{L}_{kmns} = \frac{1}{N}\sum_{i=1}^{N}\min_{\mathbf{c}_k} d(\mathbf{h}^{(i)}, \mathbf{c}_k). \tag{1}$$

In this expression, N is the number of training patches, $d(\cdot,\cdot)$ is the L_2 distance between two vectors, $\mathbf{c}_k$ is the centroid of the cluster k, $\mathbf{h}^{(i)} = g\big(f_\theta(\mathbf{x}^{(i)})\big)$ are the principal components of the feature representations of the i^{th} input training patch $\mathbf{x}^{(i)}$, $f_\theta(\cdot)$ is the CNN that produces the feature representation, and $g(\cdot)$ computes principal component analysis (PCA). Note that we perform PCA (Wold *et al.*, 1987) on the feature representations before clustering in order to use only the first few principal components for manageable computational complexity. Also note that the CNN remains fixed without being optimized in this step.

Next, we optimize the CNN to learn the feature representations clustered by k-means. The CNN is trained in a supervised manner by the supervision of the pseudo-labels, not the annotations, where the assignment indices from the result of the k-means clustering become the pseudo-labels. A cross-entropy loss, which is a standard choice for the classification task in SDL, is used for the optimization.

To align the lengths of the feature representation and the pseudo-label to K, we append a single fully connected (FC) layer with a softmax at the end of the CNN, depicted as FC1 in Figure 4. The CNN appended by FC1 becomes an end-to-end learning model.

The clustering objective is depicted as:

$$\mathcal{L}_{cls} = \frac{1}{N}\sum_{i=1}^{N} CE\{\tilde{f}_\theta(\mathbf{x}^{(i)}), \hat{\mathbf{y}}^{(i)}\}, \tag{2}$$

where $CE(\mathbf{z},\mathbf{y}) = -\sum_k y_k \log(z_k)$ is the cross-entropy loss of a single datapoint, $\hat{\mathbf{y}}^{(i)} \in \{0,1\}^K$ is the one-hot encoded pseudo-label of $\mathbf{x}^{(i)}$, and $\tilde{f}_\theta(\cdot)$ is the FC1-appended CNN that produces the pseudo-label prediction. The entire set of the pseudo-labels is changed each time when a new clustering result is obtained. We randomly initialize the weights of FC1, which aligns the representation of the CNN to the pseudo-labels, for each new update of the pseudo-label set.

Classification objective

The classification objective enforces consistency of predictions with regard to the given classes in the partially available annotated data. Using available annotated data, we train the model in a supervised manner with respect to the given classes, anticipating that the model learns the feature representations to compact each cluster in terms of the annotated data. The learned representations are reflected in updating the clustering structure, in such a way that the structure converges with respect to the given class distribution. Note that the class indices matter in this step. After removing FC1 from the CNN, we append another FC layer with softmax, called FC2, at the same place, to learn the class prediction using the cross-entropy loss. The CNN appended by FC2 also becomes an end-to-end learning model.

The classification objective is depicted as:

$$\mathcal{L}_{sup} = \frac{1}{L}\sum_{i=1}^{L} CE\{\ddot{f}_\theta(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}\}, \tag{3}$$

where $L \le N$ represents the number of annotated data, C represents the number of classes to predict, $\mathbf{y}^{(i)} \in \{0, 1\}^C$ represents the annotation of $\mathbf{x}^{(i)}$, and $\ddot{f}_\theta(\cdot)$ represents the FC2-appended CNN that produces the class prediction.

Training procedure

The combined optimization leveraging both the clustering objective and the classification objective in the single CNN constitutes a novel semi-SDL method. The training procedure consists of three stages: (1) create pseudo-labels using k-means; (2) optimize the model using the clustering objective; and (3) optimize the model using the classification objective. The iteration of the stages from (1) to (3) optimizes the CNN. Figure 4, Algorithm 1, and Algorithm 2 illustrate the procedures.

(1) Create pseudo-labels using k*-means*

The CNN provides the feature representations by processing all training patches. These principal components of the feature representations processed by PCA are clustered to K clusters by k-means as shown in Equation (1). The cluster index of each patch becomes a pseudo-label. This stage is done when each patch in the training set has its cluster index that implies the clustering structure. The CNN processes the patches but is not optimized in this stage.

(2) Optimize the model using the clustering objective

This stage aims to optimize the CNN under the supervision of the pseudo-labels. We first construct the pairs consisting of the patch and the pseudo-label. The pseudo-labels should be cluster-balanced to avoid the trivial solutions of the k-means (Yang *et al.*, 2017). To enforce this balance, we sample pairs from each cluster up to the average number of patches per cluster. Replacement is tolerated if the cluster does not have enough pairs in it with respect to this average number of patches per cluster. We append FC1 and train the CNN in an end-to-end manner with these uniformly sampled pairs, where FC1 has weights which maps the feature representations before PCA to K clusters, and zero bias as depicted in Equation (2). The CNN is optimized by the gradients that backpropagates via FC1. Note that FC1 is not optimized as the cluster indices are randomly changeable. Instead we initialize the parameters in FC1.

(3) Optimize the model using the classification objective

This stage aims to learn by the supervision of a few available class-wise annotations (three classes in our case study). FC1 is removed from the end of the CNN, and FC2 with zero bias and the weight that maps the feature representations to given labeled classes is appended. Note that we keep the parameters of FC2 from the previous turn to maintain consistency of the class prediction.

This provides another end-to-end learning model that is supervised by the annotation of three classes as shown in Equation (3). The model including the CNN and FC2 is updated with gradient decent. The prediction accuracies of the training set is measured after the optimization. For the next iteration, we remove FC2 after exporting the weight values and repeat the stage **(1)**.

Algorithm 1 Create pseudo-labels using k-means

```
Input: training patches $\mathcal{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$
Output: pseudo-labels $\hat{\mathcal{Y}} = \{\hat{\mathbf{y}}^{(i)}\}_{i=1}^{N}$
Procedure:
  while $i \neq N$ do
    Process $\mathbf{x}^{(i)}$ to the CNN $f_\theta$
    Reduce dimension of $f_\theta(\mathbf{x}^{(i)})$ using PCA and store
  end while
  Cluster the stored feature representations using $k$-means
  Create pseudo-label $\hat{\mathbf{y}}^{(i)}$ using the cluster assignment of $\mathbf{x}^{(i)}$
```

Algorithm 2 Optimize the model by alternating two objectives

```
Input: $\mathcal{X}, \hat{\mathcal{Y}}$ from Algorithm 1 and class annotation $\{\mathbf{y}^{(i)}\}_{i=1}^{L \le N}$
Procedure:
  Sample the same number of $(\mathbf{x}^{(i)}, \hat{\mathbf{y}}^{(i)})$ pairs w.r.t pseudo-label
  Append randomly initialized FC1 at the end of CNN: $\tilde{f}_\theta$
  while $i \neq N$ do
    Process $\mathbf{x}^{(i)}$ to the CNN $\tilde{f}_\theta$
    Compute loss with (??) and update $\Theta$ with gradient descent except FC1
    if $\mathbf{y}^{(i)}$ exists then
      Replace FC1 to FC2: $\ddot{f}_\theta$
      Process $\mathbf{x}^{(i)}$ to the CNN $\ddot{f}_\theta$
      Compute loss with (??) and update $\Theta$ with gradient descent
    end if
  end while
```

Organizing training data for semi-SDL

The semi-supervised method we propose exploits both the data structure in the entire set of training patches as well as in a few annotated patches.

Under the assumption that the total number of the patches is fixed, data organization for the method is characterized by the annotation ratio, which indicates the ratio of the annotated patches to the entire set of training patches. We set the total number of the training patches to 32766, and the total number of the test patches to 18012 as depicted in Table 1.

Annotation ratio

To construct the training data for the proposed method, we introduce the annotation ratio, which measures the ratio of the number of annotated patches to the number of the entire set of training patches. Four ratios are studied, namely, 1.000, 0.100, 0.050, and 0.025, where the annotation ratio of 1.000 represents full supervision. Table 2 illustrates the number of annotated and unannotated patches for each annotation ratio, where the number of unannotated patches is the same over the classes as we annotate patches according to the annotation ratio from the undersampled training patches. We refer this as unannotated-balanced (U-Ba), since the unannotated part is class-balanced. Figure 5 depicts the t-SNE plots of U-Ba with the annotation ratio of 0.100 case.

Table 2. The number of training patches for U-Ba case with respect to the classes BG, SE, and OT, and the annotation ratio.

Anno. ratio	**1.000**		**0.100**		**0.050**		**0.025**	
U-Ba	**Anno.**	**Unanno.**	**Anno.**	**Unanno.**	**Anno.**	**Unanno.**	**Anno.**	**Unanno.**
BG	10 922	0	1092	9830	546	10 376	273	10 649
OT	10 922	0	1092	9830	546	10 376	273	10 649
SE	10 922	0	1092	9830	546	10 376	273	10 649
Total	32 766	0	3276	29 490	1638	31 128	819	31 947
	32 766		32 766		32 766		32 766	

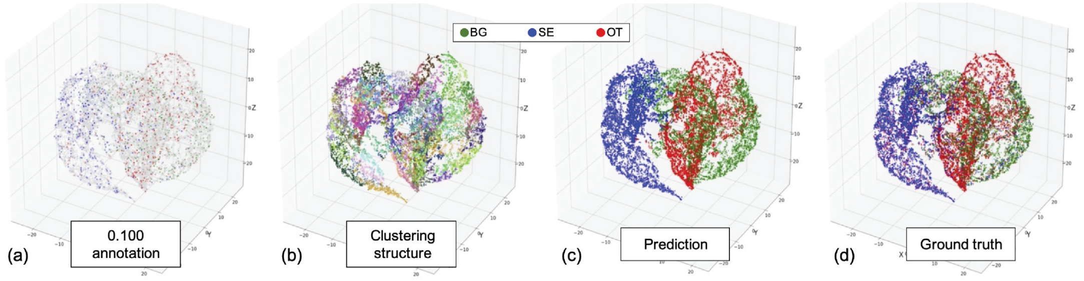


Figure 5. Three-dimensional t-SNE plots of the training patches (U-Ba, 0.100). (a) The distribution of training patches in an arbitrary space. Colored points represents the annotated patches, while gray points are unannotated ones. (b) Clustering structure of 81 clusters. The color differentiates the cluster assignment. (c) Class prediction. (d) The ground truth of the prediction.

Preserving class imbalance in unannotated part

It is important for the deep learning model to have a class-balanced training dataset since the imbalance of the data may cause bias that harms the generalization of the model prediction (Goodfellow *et al.*, 2016). To comply with this rule of thumb, we set the annotated part of the data to be class-balanced. However, when it comes to the unannotated part of the data, the rule of thumb is not applicable since the annotations are not accessible to know whether it is balanced or not.

The impact of the class imbalance in the unannotated part should be independently considered as this may potentially affect the performance of the proposed method. From our extracted patches, we observe the severe class imbalance. As shown in Table 1, 97.81% of the patches belong to the BG class.

To measure the robustness of the proposed method against the class imbalance in the unannotated part of the data, we institute a new setting referring to as unannotated-imbalanced (U-Im) in addition to U-Ba, where U-Im simulates the intrinsic class distribution before undersampling patches. Table 3 specifies the number of patches for the U-Im case. Note that the annotated part and the total number of patches are the same for those two cases.

Experiments

The purpose of the experiment on our SE case study is to explore the robustness of the proposed method in the semi-supervised learning environment that exploits limited annotations and, at the same time, the contribution of the unannotated data. In the experiments, we observe the prediction accuracy of the proposed method with different settings of the training set in terms of the annotation ratio and the unannotated data.

Unannotated data

Two settings for the unannotated part, U-Ba and U-Im, are suggested above. In parallel, to measure the lowerbound performance of the proposed model in terms of the unannotated data, we construct additional training sets that use only the annotated part of the data which is class-balanced, referred to as annotated only (AO). The number of patches over the classes is given in Tables 2 and 3. For example, with the annotation ratio of 0.025, the training set for AO case consists of 819 annotated patches without any unannotated patches. An annotation ratio of 1.000 is included in order to estimate the upperbound of the proposed method, where the model exploits full supervision of the annotations, while simultaneously learning the structure with the clustering objective.

Model description

We create our own CNN based on the architecture of VGG-16, but modify a few points including the input layer to utilize the four-channel patches in our CNN architecture.

The VGG-16 can be broadly divided into two parts, a feature extractor and a classifier. The feature extractor consists of in total 18 layers, 5 max-pooling layers with 2 × 2 kernels and 13 convolution layers with 3 × 3 filters, where the max-pooling layers are located in the 3^{rd}, 6^{th}, 10^{th}, 14^{th}, and 18^{th} layers. The remaining layers are convolution layers. Each convolutional layer is followed by batch normalization (Ioffe and Szegedy, 2015) and a rectified linear unit (ReLU) activation (Nair and Hinton, 2010). Based on the location of the pooling layer, the number of filters for each convolution layer

Table 3. The number of training patches for U-Im case with respect to the class and the annotation ratio.

Anno. ratio	**1.000**		**0.100**		**0.050**		**0.025**	
U-Im	**Anno.**	**Unanno.**	**Anno.**	**Unanno.**	**Anno.**	**Unanno.**	**Anno.**	**Unanno.**
BG	10 922	0	1092	28 845	546	30 446	273	31 248
OT	10 922	0	1092	383	546	405	273	415
SE	10 922	0	1092	262	546	277	273	284
Total	32 766	0	3276	29 490	1638	31 128	819	31 947
	32 766		32 766		32 766		32 766	

The unannotated part is shared according to their intrinsic distribution such that BG, OT, and SE classes occupy 97.81%, 1.30%, and 0.89%, respectively.

varies in 5 steps, where the first 2 layers have 64, the 4^{th} and the 5^{th} layers have 128, the layers from the 7^{th} to the 9^{th} have 256, and the layers from the 11^{th} to the 13^{th} and the 15^{th} to the 17^{th} have 512 filters.

We leverage the feature extractor part of VGG-16 with a modification of the input layer. Due to 5 max-pooling layers with with 2 × 2 kernels, the model reduces the dimension of the input patches to $1/2^5$, and the feature representations before the classifier have the vectorial form of $(1 \times 1 \times 512)$ that can be input to the classifier without flattening.

The classifier of VGG-16 has three FC layers with ReLU activation. To remove the effect from ReLU before *k*-means clustering, the last ReLU activation is discarded when the output of the classifier is supposed to be used for PCA. For regularization, dropout ($p = 0.5$) (Srivastava *et al.*, 2014) is performed after the first and second activation function in the classifier. The number of neurons for each layer is 4096, 4096, and 128, respectively. The outcome for the echosounder patch is set to a vector of length 128 considering the balance between the computational complexity and the available computing resources.

Training configuration

The model is trained by the use of mini-batch training, where the batch size is set to 32. The Adam optimizer (Kingma and Ba, 2015) with learning rate 3×10^{-5}, beta (0.9, 0.999), and weight decay 10^{-5} is applied for the all experiments. The three-stage training shown in Figure 4 is iterated 1000 times for all experiments, applying early stopping (Prechelt, 1998) on the condition that the accuracy is not improved for 100 times. We choose the first 32 principal components in Equation (1) as they capture most of the variance of the data. The training procedure for the proposed method is shared for all experiments. As discussed in the study of Caron *et al.* (2018), the choice of the number of the clusters *K* does not have a significant impact on the performance if we cluster the feature representations with a sufficiently large number of clusters compared to the number of classes. We have tested a set of different *K*s, and choose *K* to be 81 considering the following reasons. (i) Classifying the patches up to $C = 3$ classes, we expect *K* to be expressed in terms of the number of classes *C*, such as $K = C^4$, expecting that each class has approximately C^3 clusters for the U-Ba case. (ii) Considering the total number of training patches $N = 32766$, the average number of patches in a cluster is approximately 400. Under the scenario of an annotation ratio of 0.025, each cluster has approximately 10 annotated patches. We tune those hyperparameters using the patches that are excluded from the training set and the test set in the undersampling process. All the codes are implemented in PyTorch (Paszke *et al.*, 2017).

Validation methods

For the validation of the proposed method, we introduce two baseline models to compare the performance. The first baseline is introduced to compare the performance of our deep learning method to a robust semi-supervised machine learning algorithm. We utilize the advanced semi-supervised support vector machine (S3VM) (Bagattini *et al.*, 2017), a statistical learning framework that is frequently used in many real-world applications. The S3VM classifier is trained based on the learned feature representations of length 128 from the proposed model using the radial basis function kernel for this non-linear classification problem.

The second baseline allows us to investigate the impact of the clustering objective in a supervised condition. The AO settings play this role. The proposed method that utilizes two objectives is compared with a common SDL model that leverages the classification objective only. The number of training patches for the AO settings depend on the annotation ratio as shown in Tables 2 and 3, where the patches are class-balanced. For the common SDL model, the entire training settings including the CNN architecture and related hyperparameters are shared with the proposed method in a supervised manner.

Results

Here, we focus mainly on the results form the class-balanced test set, as it demonstrates an impartial performance comparison that is not affected by the large class-imbalance.

For the class-balanced test case, the prediction accuracies for our SE case study within acoustic target classification as well as the F1 scores are presented in Table 4, where the best results are highlighted in bold. Overall, for the semi-supervised settings such as U-Im and U-Ba, the proposed model outperforms the semi-supervised benchmark S3VM (Bagattini *et al.*, 2017), and for the supervised settings referred to as AO, the proposed model achieves improved or comparable prediction performance compared to the standard SDL models over the entire set of annotation ratios. Figure 6 visualizes the prediction of the proposed method using t-SNE plots (Van der Maaten and Hinton, 2008).

Supervised case

Comparing the proposed method (ours) with the standard SDL under the AO setting with an annotation ratio of 1.000, ours (accuracy

Table 4. Prediction accuracies and F1 scores for the class-balanced test set.

Class-bal.	Accuracy						F1 score (three classes, macro averaging)					
test set	Semi-supervised				Supervised		Semi-supervised				Supervised	
Annotation	U-Im		U-Ba		AO		U-Im		U-Ba		AO	
ratio	Ours	S3VM	Ours	S3VM	Ours	SDL	Ours	S3VM	Ours	S3VM	Ours	SDL
1.000					**0.8202**	0.8000					**0.8190**	0.7966
0.100	**0.7814**	0.7341	**0.7896**	0.7340	**0.7531**	0.7496	**0.7794**	0.7313	**0.7872**	0.7313	**0.7481**	0.7462
0.050	**0.7484**	0.6668	**0.7694**	0.6668	0.6899	**0.6909**	**0.7447**	0.6653	**0.7666**	0.6654	0.6840	**0.6886**
0.025	**0.7364**	0.5838	**0.7159**	0.5827	**0.6495**	0.6108	**0.7326**	0.5774	**0.7153**	0.5765	**0.6468**	0.6109

S3VM (Bagattini *et al.*, 2017) and the standard SDL models are introduced as the benchmarks. The prediction accuracies and F1 scores of the test set are presented with respect to the settings of the training set.
Bold values denote statistical significance at the p < 0.05 level.

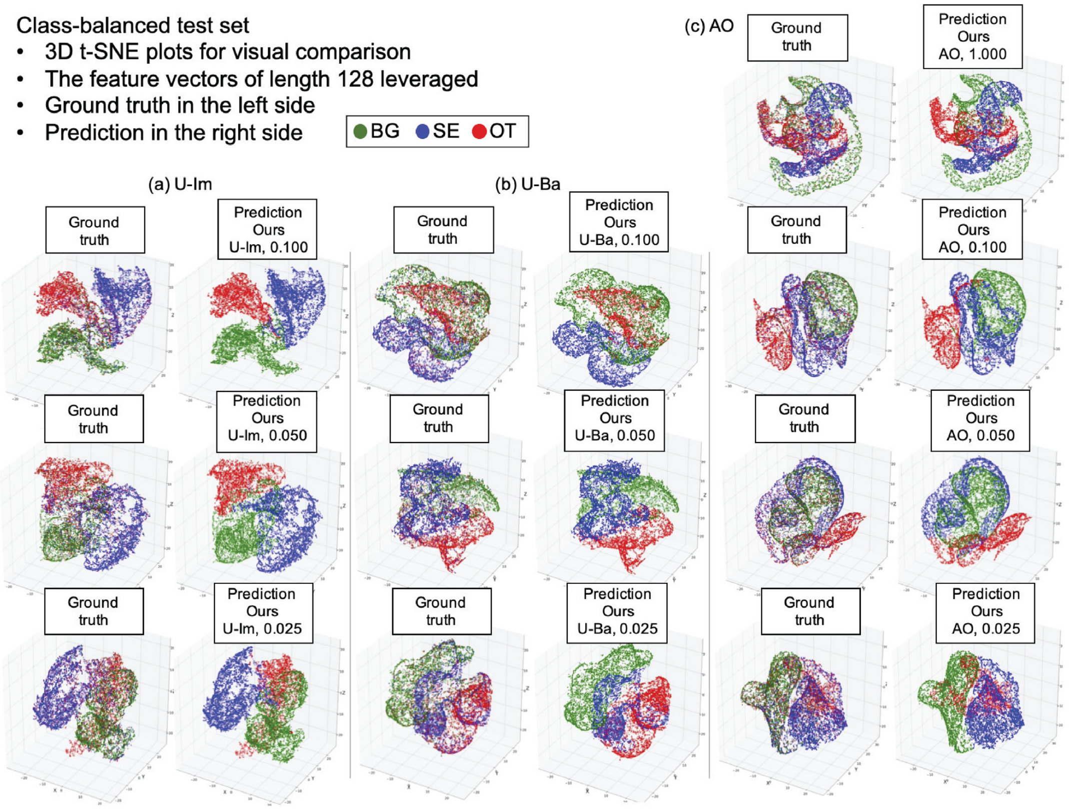


Figure 6. t-SNE plots for visual comparison (class-balanced test set). The feature vectors of the CNN for each setting are compressed for the three-dimensional plot. (a) U-Im, (b) U-Ba, and (c) AO. Less difference between the ground truth and prediction is observed from the higher annotation ratio.

0.8202) outperforms the standard SDL (accuracy 0.8000) by 2.02 percentage points. This trend is consistent also with other annotation ratios.

These results validate that the proposed method leveraging the unsupervised clustering objective improves the prediction performance over common SDL. We argue that the alternating optimization of the two proposed objectives leads the model to understand more about the global data distribution, and this contributes to creating improved decision boundaries compared to the traditional supervised learning approach that learns to mimic the given class attributes in the training set.

Annotation ratio

Throughout the cases, we observe as a tendency that the prediction accuracy increases as the annotation ratio increases. Interestingly, there is only 1.86 percentage points difference in accuracy between the proposed model with U-Im with 0.100 annotation ratio (U-Im, accuracy 0.7814) and the standard SDL with 1.000 annotation ratio (SDL, accuracy 0.8000), where the proposed method leverages a tenth of the annotated data against the standard SDL setting. The proposed method also outperforms the same annotation ratio (0.100) case of the standard SDL (accuracy 0.7496) by 3.18 percentage points.

The results indicate that the proposed method can effectively exploit a small amount of annotated data, and, to a certain extent, approximate the decision boundaries that are achieved by the fully SDL. We argue that, in the proposed method, the annotated data are leveraged by two different objectives respectively, which facilitate the interconnection of the two objectives in order to make good use of the annotated data. In this process, the unannotated data in a cluster gradually share the annotations that originate from the annotated data in the same cluster or the clusters nearby located, and eventually, the entire data in the same cluster have the same class prediction.

Class imbalance in unannotated data

In our method, the utilization of the unannotated data, found in the U-Im and U-Ba cases, considerably improves the prediction performance compared to the AO case under the same annotation ratio. In particular, the U-Im is comparable to the U-Ba setting. This includes the case where we in the U-Im setting (accuracy 0.7364) achieve 2.05 percentage points higher accuracy compared to the U-Ba setting (0.7159) with 0.025 annotation ratio. Those are promising results as a severe class imbalance is observed in the unannotated data for the U-Im case.

Confusion matrices

Figure 7 depicts the confusion matrices of the class-balanced test set, with respect to the annotation ratio and the unannotated part of the training set. For each matrix, the class BG is represented by the first row/column, the class SE is represented by the second row/column, and the class OT can be found in the third row/column. Each true class consists of one row and the probabilities of each row sums to one.

When comparing the diagonal components of the two confusion matrices for the semi-supervised cases, the proposed method can be seen to outperform the benchmark for all the classes and settings except two cases for the OT class in the U-Ba setting, where the accuracies are comparable (ours: 0.7840, S3VM: 0.7916 with the annotation ratio of 0.100, and ours: 0.7350, S3VM: 0.7465 with the annotation ratio of 0.025). Also, the degree of improvement is greater in the SE and BG classes than in the OT class. We believe that the reason for this is that the training patches in the SE and BG classes are more uniform than the ones in the OT class, which capture the backscattered response from diverse fish species when collected, and that deep clustering takes advantage of the uniformity when investigating the structure of the data.

We observe that the BG class achieves higher accuracy than the other classes, probably since the backscattering intensities in the BG patches are considerably more uniform, mostly having the lowest intensity. The SE class shows the lowest accuracy among the classes (e.g. 0.6755, U-Im with annotation ratio of 0.100), resulting in a higher false-negative rate (0.3245) and lower false-positive rate (0.1604).

This means that the predicted amount of SE will be a conservative estimate as the SE patches are frequently misclassified to other classes but the patches in the other classes are rarely misclassified to the SE class. We do not observe a tendency for any bias towards one class over the other for the misclassified SE patches.

The proposed method achieves more consistent performance against the variation of the annotation ratios compared to the benchmark in the semi-supervised cases. We argue that the proposed method is robust even the available annotated data are extremely few, as it approximates the relatively accurate decision boundary for the prediction by understanding the global distribution of the data, along with learning how to effectively exploit the available annotated data.

Class-imbalanced test set

For the class-imbalanced test case, the prediction accuracies and the F1 scores are presented in Table 5, where the best result is highlighted in bold. Note that severe class imbalance causes bias in the result to a certain degree, where 97.61% of the test patches belong to the BG class as depicted in Table A1 in the Appendix. Overall, we observe the similar tendency that we discover from Table 4, where the proposed method outperforms the semi-supervised benchmark. Confusion matrices for the class-imbalanced test case is shown in Figure A1 in the Appendix.

For the class-imbalanced test case, the prediction accuracies and the F1 scores are presented in Table 5, where the best result is highlighted in bold. Note that the severe class imbalance causes a bias in the result as 97.61% of the test patches belong to the BG class as depicted in Table A1. Overall, we observe a similar tendency to what we discover from Table 4, where the proposed method outperforms the semi-supervised benchmark. Confusion matrices for the class-imbalanced test case are shown in Figure A1 in the Appendix.

Conclusion

In this paper, we proposed a novel semi-SDL method for acoustic target classification, which (ii) takes advantage of the power of deep learning, (ii) is trainable end-to-end in both semi-supervised and fully supervised manners, (iii) exploits the underlying structure of the training data regardless of the annotation, (iv) is robust against the class imbalance of the unannotated part of the data, and (v) achieves results that outperform or are comparable with other methods including a common SDL model. We have also investigated the performance through extensive experiments to evaluate the robustness of the method using rigorous criteria and compare the results with the advanced machine learning benchmark model. In addition, we have established a data organization process for semi-supervised learning to tackle the challenge of class imbalance. Overall, the promising results imply that the proposed method including the data organization process can be broadly applied to the severely class-imbalanced data with limited annotations, which are often found in the real world. To the best of our knowledge, this is the first semi-SDL paper in acoustic target classification.

In future work, we intend to explore other types of deep neural networks architectures beside the VGG-16 network. It would also be of interest to study other types of acoustic target classification problems. As a further example of future work, we intend to extend our method in order to categorize a single intensity of the multi-frequency echosounder data. This is known as pixel-level semantic segmentation, which potentially can contribute to

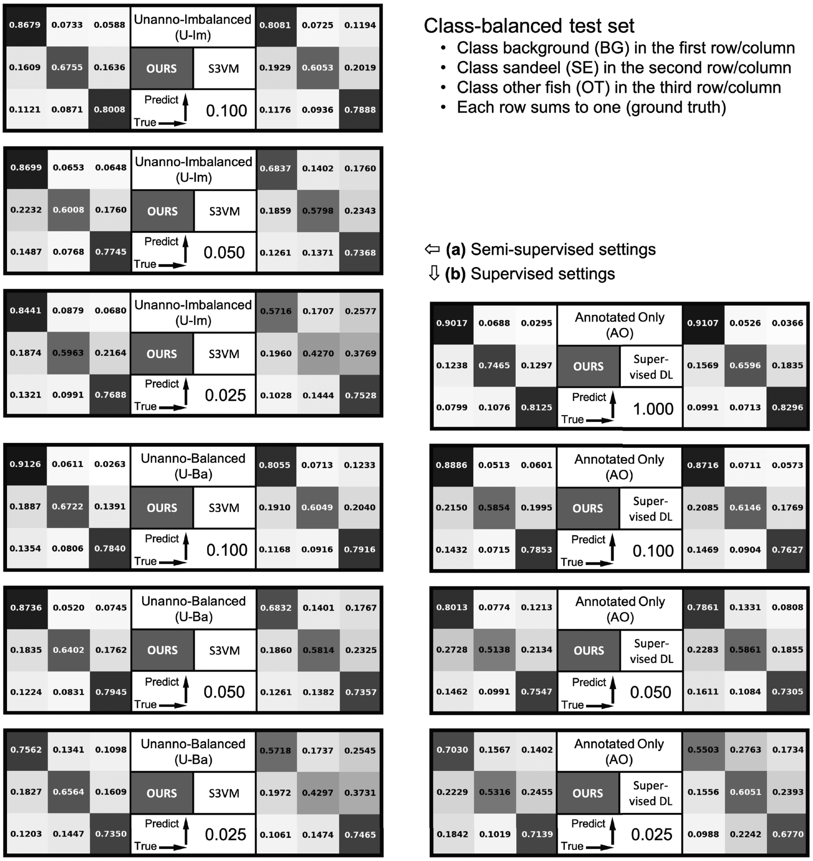


Figure 7. Confusion matrices (3 × 3) of the class-balanced test set. Each diagonal element of each matrix indicates the ratio of the number of correctly predicted patches in the corresponding class to the number of patches in the true class. (a) The matrices in the left column represent the semi-supervised settings (U-Im and U-Ba). (b) The matrices the right column represents the supervised settings (AO). The number next to the arrows between two matrices indicates the annotation ratio.

Table 5. Prediction accuracies and F1 scores for the class-imbalanced test set.

Class-imbal.	**Accuracy**						**F1 score (three classes, weighted averaging)**					
tests set	**Semi-supervised**				**Supervised**		**Semi-supervised**				**Supervised**	
Annotation	**U-Im**		**U-Ba**		**AO**		**U-Im**		**U-Ba**		**AO**	
ratio	**Ours**	**S3VM**	**Ours**	**S3VM**	**Ours**	**SDL**	**Ours**	**S3VM**	**Ours**	**S3VM**	**Ours**	**SDL**
1.000					0.9026	**0.9098**					0.9350	**0.9382**
0.100	**0.8621**	0.8013	**0.9099**	0.7996	**0.8809**	0.8653	**0.9095**	0.8713	**0.9392**	0.8702	**0.9202**	0.9112
0.050	**0.8617**	0.6860	**0.8676**	0.6858	**0.7944**	0.7789	**0.9088**	0.7952	**0.9121**	0.7950	**0.8672**	0.8580
0.025	**0.8412**	0.5628	**0.7498**	0.5623	**0.6988**	0.5436	**0.8969**	0.7018	**0.8390**	0.7012	**0.8044**	0.6863

S3VM (Bagattini *et al.*, 2017) and the standard SDL models are introduced as the benchmarks. The best result is highlighted in bold.

a more precise estimation of biomass or fish abundance. We will also investigate the proposed method in other domains of structured data analysis to assess whether our method generalizes to other applications. We are also interested in developing the neural networks that process missing data using internal computational mechanisms, as the missing ping is commonly found during data acquisition phase and can deteriorate the robustness of the analysis.

Supplementary Data

Supplementary material is available at the *ICESJMS* online version of the manuscript.

Data Availability Statement

The data underlying this article were provided by the Institute of Marine Research of Norway by permission. Data will be shared on request to the corresponding author with permission of the Institute of Marine Research of Norway.

Acknowledgements

This work was financially supported by the Research Council of Norway (RCN), through its project for Ubiquitous Cognitive Computer Vision for Marine Services (COGMAR, grant no. 270966) and its Center for Research-based Innovation in Marine Acoustic Abundance Estimation and Backscatter Classification (CRIMAC, grant no. 309512).

REFERENCES

Bagattini, F., Cappanera, P., and Schoen, F. 2017. Lagrangean-based combinatorial optimization for large-scale S3VMs. IEEE Transactions on Neural Networks and Learning Systems, 29: 4426–4435.

Biernacki, C., Celeux, G., and Govaert, G. 2000. Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22: 719–725.

Brautaset, O., Waldeland, A. U., Johnsen, E., Malde, K., Eikvil, L., Salberg, A.-B., and Handegard, N. O. 2020. Acoustic classification in multifrequency echosounder data using deep convolutional neural networks. ICES Journal of Marine Science. 77: 1391–1400.

Buda, M., Maki, A., and Mazurowski, M. A. 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 106: 249–259.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. European Conference on Computer Vision (ECCV), pp. 132–149.

Chapelle, O., Scholkopf, B., and Zien, A. 2009. Semi-supervised learning. IEEE Transactions on Neural Networks, 20: 542–542.

Daan, N., Bromley, P., Hislop, J., and Nielsen, N. 1990. Ecology of north sea fish. Netherlands Journal of Sea Research, 26: 343–386.

Frederiksen, M., Furness, R. W., and Wanless, S. 2007. Regional variation in the role of bottom-up and top-down processes in controlling sandeel abundance in the north sea. Marine Ecology Progress Series, 337: 279–286.

Furness, R. W. 2002. Management implications of interactions between fisheries and sandeel-dependent seabirds and seals in the north sea. ICES Journal of Marine Science, 59: 261–269.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. 2016. Deep Learning, vol. 1. MIT Press, Cambridge.

Handegard, N. O., and Tjøstheim, D. 2009. The sampling volume of trawl and acoustics: estimating availability probabilities from observations of tracked individual fish. Canadian Journal of Fisheries and Aquatic Sciences, 66: 425–437.

ICES 2017. Report of the Benchmark Workshop on Sandeel (WKSand 2016), 31 October - 4 November 2016, Bergen, Norway. International Council for the Exploration of the Sea (ICES). ICES Document CM 2016/ACOM:33. 319 pp.

Ioffe, S. and Szegedy, C. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning (ICML), pp. 448–456.

Jabi, M., Pedersoli, M., Mitiche, A., and Ayed, I. B. 2019. Deep clustering: on the link between discriminative models and k-means. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43: 1887–1896.

Johnsen, E., Pedersen, R., and Ona, E. 2009. Size-dependent frequency response of sandeel schools. ICES Journal of Marine Science, 66: 1100–1105.

Johnsen, E., Rieucau, G., Ona, E., and Skaret, G. 2017. Collective structures anchor massive schools of lesser sandeel to the seabed, increasing vulnerability to fishery. Marine Ecology Progress Series, 573: 229–236.

Kampffmeyer, M., Løkse, S., Bianchi, F. M., Livi, L., Salberg, A.-B., and Jenssen, R. 2019. Deep divergence-based approach to clustering. Neural Networks, 113: 91–101.

Kingma, D. P. and Ba, J. 2015. Adam: a method for stochastic optimization. International Conference on Learning Representations (ICLR).

Kloser, R., Ryan, T., Sakov, P., Williams, A., and Koslow, J. 2002. Species identification in deep water using multiple acoustic frequencies. Canadian Journal of Fisheries and Aquatic Sciences, 59: 1065–1077.

Korneliussen, R. J. 2018. Acoustic target classification. ICES Cooperative Research Report No. 344. 104 pp. International Council for the Exploration of the Sea (ICES).

Korneliussen, R. J., Heggelund, Y., Macaulay, G. J., Patel, D., Johnsen, E., and Eliassen, I. K. 2016. Acoustic identification of marine species using a feature library. Methods in Oceanography, 17: 187–205.

Korneliussen, R. J., and Ona, E. 2003. Synthetic echograms generated from the relative frequency response. ICES Journal of Marine Science, 60: 636–640.

Long, J., Shelhamer, E., and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

MacLennan, D. N., Fernandes, P. G., and Dalen, J. 2002. A consistent approach to definitions and symbols in fisheries acoustics. ICES Journal of Marine Science, 59: 365–369.

MacLennan, D. N., and Simmonds, E. J. 2013. Fisheries Acoustics, vol. 5. Springer Science & Business Media. Berlin, Germany.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. International Conference on Machine Learning (ICML), p. 807–814.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z. *et al.* 2017. Automatic differentiation in pytorch. NIPS 2017 Workshop on Autodiff.

Prechelt, L. 1998. Early stopping-but when?In Neural Networks: Tricks of the Trade, pp. 55–69. Springer. New York City, USA.

Raitt, D. 1934. A preliminary account of the sandeels of scottish waters. ICES Journal of Marine Science, 9: 365–372.

Reid, D. G. 2000. Report on echo trace classification. ICES Cooperative Research Report No. 238. International Council for the Exploration of the Sea (ICES).

Rezvanifar, A., Marques, T. P., Cote, M., Albu, A. B., Slonimer, A., Tolhurst, T., Ersahin, K. *et al.* 2019. A deep learning-based framework for the detection of schools of herring in echograms. Tackling Climate Change with Machine Learning Workshop at Advances in Neural Information Processing Systems (NeurIPS).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15: 1929–1958.

Van der Maaten, L., and Hinton, G. 2008. Visualizing data using t-SNE. Journal of Machine Learning Research, 9: 2579–2605.

Wold, S., Esbensen, K., and Geladi, P. 1987. Principal component analysis. Chemometrics and intelligent laboratory systems, 2: 37–52.

Yang, B., Fu, X., Sidiropoulos, N. D., and Hong, M. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. International Conference on Machine Learning (ICML), pp. 3861–3870.

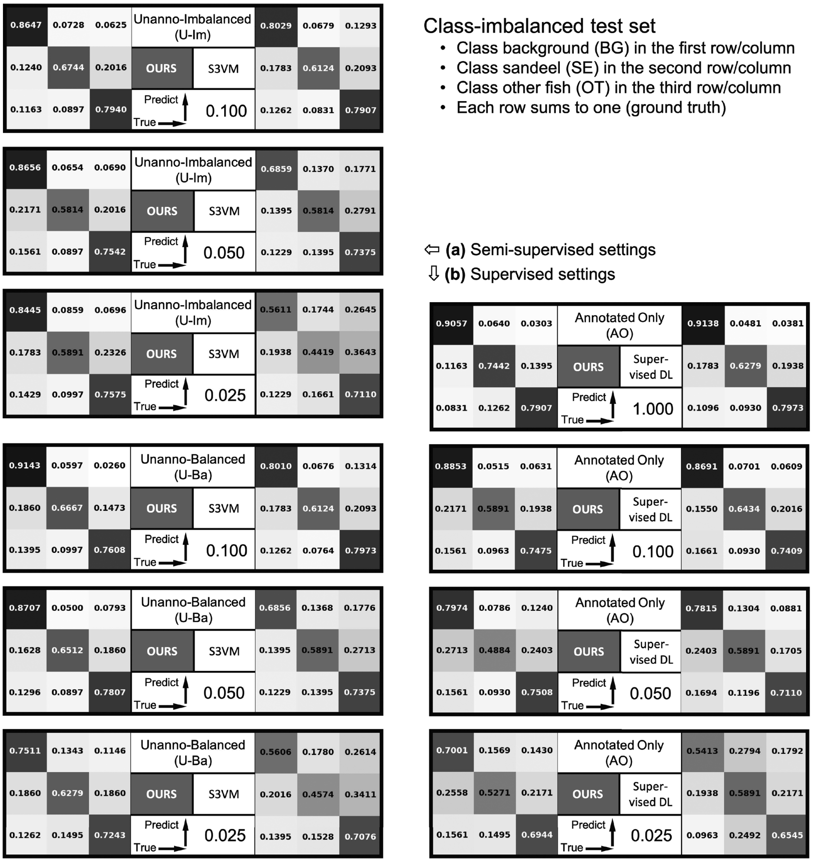


Figure A1. Confusion matrices (3 × 3) of the class-imbalanced test set, where the first row/column indicates BG, the second one is OT, and the third one is SE.

Table A1. The number of test patches sampled in a way to preserve the intrinsic class imbalance from the test echosounder data (2018–2019).

Year	**Test set (2018–2019)**	
Class	**Extracted patches (percentage)**	**Sampled by intrinsic distr.**
BG	816 726 (97.61)	17 582
OT	6004 (0.72)	129
SE	13 984 (1.67)	301
Total	836 714 (100.00)	18 012

Appendix

Deep learning terminologies

Epoch indicates that the model has performed a single pass over the entire training set.

Loss function is a measure of how good a model is performing for a specific task. A high value of the loss function indicates poor model performance. In order to improve the performance of the model for the given task, the loss is minimized.

One-hot encoding is a method to quantify categorical data by producing a vector with length equal to the number of categories in the data set. If a data point belongs to the i^{th} category then all elements of this vector are assigned the value 0 except for the i^{th} component which is assigned a value of 1.

Softmax function is a generalization of the logistic function to multiple dimensions. It is used in multi-class classification and is often used as the last activation function of a neural network to normalize the output of a network to a probability distribution over predicted output classes.

Cross-entropy loss measures the performance of a classification model whose output is a probability value between 0 and 1. The cross-entropy loss increases as the predicted probability diverges from the actual label. The ideal model would have the loss of 0, where an outcome of the model has a form of a one-hot vector.

End-to-end learning model refers to training a possibly complex learning system represented by a single model that represents the complete target system, bypassing the intermediate layers usually present in traditional pipeline designs.

Handling Editor: Cigdem Beyan

10 | Paper II

Deep Semisupervised Semantic Segmentation in Multifrequency Echosounder Data

Changkyu Choi, Michael Kampffmeyer, *Member, IEEE*, Nils Olav Handegard, *Member, IEEE*, Arnt-Børre Salberg, *Member, IEEE*, and Robert Jenssen, *Senior Member, IEEE*

***Abstract*—Multifrequency echosounder data can provide a broad understanding of the underwater environment in a noninvasive manner. The analysis of echosounder data is, hence, a topic of great importance for the marine ecosystem. Semantic segmentation, a deep learning-based analysis method predicting the class attribute of each acoustic intensity, has recently been in the spotlight of the fisheries and aquatic industry since its result can be used to estimate the abundance of marine organisms. However, a fundamental problem with current methods is the massive reliance on the availability of large amounts of annotated training data, which can only be acquired through expensive handcrafted annotation processes, making such approaches unrealistic in practice. As a solution to this challenge, we propose a novel approach, where we leverage a small amount of annotated data (supervised deep learning) and a large amount of readily available unannotated data (unsupervised learning), yielding a new data-efficient and accurate *semisupervised* semantic segmentation method, all embodied into a single end-to-end trainable convolutional neural network architecture. Our method is evaluated on representative data from a sandeel survey in the North Sea conducted by the Norwegian Institute of Marine Research. The rigorous experiments validate that our method achieves comparable results utilizing only 40% of the annotated data on which the supervised method is trained, by leveraging unannotated data.**

***Index Terms*—Acoustic target classification, convolutional neural networks, deep clustering, deep learning, marine acoustics, multifrequency echosounder data, semisupervised semantic segmentation.**

I. Introduction

SEMANTIC segmentation is one of the fundamental computer vision tasks, where the aim is to assign each image pixel to a semantic class [1], [2], [3]. When analyzing echosounder data, the aim is to assign an observed acoustic backscattering intensity to one of several given acoustic classes, often referred to as acoustic target classification [4], [5], [6], [7]. In practice, semantic segmentation of the echosounder data is still a manual and heuristic process, which is rather vulnerable to human error and bias. It is also expensive in terms of cost and time [8].

There are a few studies that intend to automate the semantic segmentation based on statistical modeling and machine learning techniques [9], [10], [11], [12], [13]. However, they are exposed to limitations such as relying heavily on handcrafted feature selection and not being able to scale well to large amounts of data. As recent echosounder technology leverages increasing numbers of frequency channels and wider bandwidth [14], automated analysis methods should therefore be scalable to cope with increased resolution and multifrequency data.

Convolutional neural networks (CNN) is a framework renowned for excelling at image segmentation tasks [15]. Recent echosounder segmentation studies introduce CNN-based segmentation methods as alternative strategies [5], [16], [17], [18], [19], where the main advantage is the capacity to learn discriminating features from the training data without requiring a handcrafted process, allowing the analysis to scale to large-sized data. Note that these methods are trained in a fully supervised manner, indicating that the network learns from fully annotated training data. The fully supervised approaches achieve good performance provided that high-quality training data and an appropriate choice for the prediction model are assured. However, it is highly challenging for the echosounder data to obtain the class annotation for each backscattering intensity pixel because this relies on the manual annotation process, which is expensive and error-prone.

Hence, a new learning scheme is required to considerably reduce the dependence on the manual annotation process while still facilitating powerful deep-learning approaches for the segmentation of the echosounder data. As a key step in this direction, we propose a novel deep *semisupervised* semantic segmentation method that efficiently uses a small amount of manually annotated data by combining it with a large amount of readily available unannotated data in the learning process [20], [21], [22].

The key concept invoked to train the semisupervised segmentation network is to alternate between two objective functions, namely, an unsupervised clustering objective and a supervised

Manuscript received 31 January 2022; revised 13 September 2022 and 21 October 2022; accepted 26 November 2022. Date of publication 1 February 2023; date of current version 14 April 2023. This work was supported in part by the Ubiquitous Cognitive Computer Vision for Marine Services (COGMAR) under Research Council of Norway (RCN) Grant 270966, in part by the Centre for Research-based Innovation Visual Intelligence under RCN Grant 309439, and in part by the Centre for Research-based Innovation in Marine Acoustic Abundance Estimation and Backscatter Classification (CRIMAC) under RCN Grant 309512. *(Corresponding author: Changkyu Choi.)*

Associate Editor: H. Zheng.

Changkyu Choi, Michael Kampffmeyer, and Robert Jenssen are with the UiT The Arctic University of Norway, 9019 Tromsø, Norway (e-mail: changkyu.choi@uit.no; michael.c.kampffmeyer@uit.no; robert.jenssen@uit.no).

Nils Olav Handegard is with the Norwegian Institute of Marine Research, 5005 Bergen, Norway (e-mail: nilsolav@hi.no).

Arnt-Børre Salberg is with the Norwegian Computing Center, 0373 Oslo, Norway (e-mail: salberg@nr.no).

The code is available at https://github.com/SFI-Visual-Intelligence/PredKlus-semisup-segmentation.

Digital Object Identifier 10.1109/JOE.2022.3226214

segmentation objective, encapsulated by a single CNN. The unsupervised clustering objective is to search the underlying structure within the training data without using the class annotation. In contrast, the supervised segmentation objective is to map the input echosounder data to the given classes presented in the available annotated data. These two objective functions alternatively optimize the single CNN and gradually integrate the underlying clustering structure to the class decision boundaries presented in the small amount of annotated training data. Our proposed method can create pixel-level prediction maps using the same CNN architecture as [5] and [23]. Still, it is data-efficient because it can significantly reduce the use of the annotated data. To the best of our knowledge, our work is the first semisupervised semantic segmentation method for multifrequency echosounder data that provides prediction maps on a pixel scale, advancing the existing semisupervised method of providing patch-scale prediction maps (see Section III-C) [22]. In addition, our proposed method is end-to-end trainable, which refers to a holistic gradient-based learning system where a formulated objective function reflects the principle of a given task without requiring extensive human intervention and prior knowledge [24].

Extensive and rigorous experiments are conducted on the multifrequency echosounder data collected at the North Sea by the Norwegian Institute of Marine Research. A severe class imbalance in the echosounder data is an ever-present source of bias that prevents training of the neural networks, where 99% of the entire acoustic backscattering intensities is occupied by the background class [5], [25]. We introduce a class-rebalancing weight to each learning objective to mitigate the bias, where the weight is calculated with respect to the model prediction without relying on the annotation.

The contributions of the article are the following.

1) To propose a novel deep semisupervised semantic segmentation method for the multifrequency echosounder data, which considerably advances the existing methods.
2) To achieve comparable results with the fully supervised segmentation method by leveraging a small amount of the annotated data in addition to unannotated data.
3) To exploit the underlying structure of the training data using unsupervised deep clustering in a semisupervised learning manner.
4) To demonstrate the innovation potential of the proposed method in a real-world test case.
5) To regulate the class imbalance based on the model prediction without leveraging the annotated part of data.
6) To operate in an end-to-end and mini-batch training scheme.

II. Background

Semantic segmentation is the process of partitioning an image into mutually exclusive subsets by assigning a class annotation to each intensity of the data, in which each subset represents a meaningful region of the original image [26]. It thereby provides a comprehensive scene description that includes object class, location, and shape. A wide range of real-world problems require semantic segmentation [27], [28], [29], [30], [31], [32], such as self-driving vehicles [33], and polyp detection [34], [35], to name a few, all depending on different types of image data.

Semantic segmentation has been considered as a challenging computer vision task due to the large distribution variance as well as the huge class imbalance among objects in the input data [25]. In recent years, however, deep learning has been rapidly advancing and has become a game-changer in many image analysis tasks including semantic segmentation. The CNN [36] is a deep learning framework that has had particular success for grid-structured data such as images. Traditional CNNs consist of convolutional layers and pooling layers, where these layers are stacked in a deep and hierarchical architecture in a particular order, providing unique properties to the analysis. For example, the weight-sharing property of the convolutional filters provides a symmetric transformation between the input space and the output space, referred to as "equivariance to translation." The pooling layers help the learned representation becoming approximately invariant to small translations of the input [15], [37]. Another advantage of the CNN is a relatively more straightforward learning process than the conventional methods, where the CNN-based models learn by minimizing a formulated objective function that reflects the strategies of a given task without requiring extensive human intervention and prior knowledge, referred to as an end-to-end learning.

CNN-based segmentation models are distinguishable through their model architecture. Their architecture consists of a downstream module that extracts the abstracted feature representations of the input data and an upstream module that reconstructs the prediction map exhibiting the class attributes of each intensity in the input data based on these extracted feature representations. Thanks to the dual architecture, those models can make class predictions on arbitrary-sized inputs [38]. Fully convolutional networks [1] and U-Net [23] are representative architectures, where the models are composed of (transposed) convolutional layers and pooling layers, and end-to-end trainable depending on their formulation of the objective functions.

A. Echosounder Data

For the sustainable management of commercially harvested marine organisms, reliable information on their abundance is essential. For example, lesser sandeel, a species of fish of interest in this study, is the primary food source in the North Sea food web thanks to its ample population [39], which are the preferred prey of a variety of predators, including marine mammals, seabirds, and piscivorous fishes [40]. Therefore, monitoring sandeel stock is critical for the sustainability of the marine ecosystem and fishery management in the North Sea. The echosounder data can contribute to estimating the abundance, leveraging the characteristics of the backscattered responses and knowledge of the target species [8]. The multifrequency echosounder data that we use in this study has been collected by multifrequency Simrad EK60 echosounder systems operating at four different frequency channels on the vessel (18, 38, 120, 200 kHz), where the vessel speed is approximately ten knots. The Norwegian Institute of

Marine Research has collected the data through the annual trawl surveys in the sandeel areas in the North Sea [41].

We leverage the data preprocessing protocol from the earlier works [5], [22], for which we share the echosounder data. For each frequency channel, a volume backscattering coefficient s_v, an average amount of backscattering intensity per cubic metre [42], is stored in the 2-D echosounder data. In the physical context, the horizontal and vertical lengths of a single backscattering coefficient are, respectively, one second and 19.2 cm based on the pulse duration of 1.024 ms with respect to a common time-range grid based on the resolution of the 200 kHz echosounder data. All the volume backscattering coefficients s_v are first converted to a decibel unit (dB re 1 m^{-1}). We set the minimum value as -75 dB re 1 m^{-1}. The coefficients less than -75 dB re 1 m^{-1} or missing coefficients are imputed to the minimum values.

For segmentation of the echousounder data, one common approach is a manual annotation method, which relies on the operators' domain expertise of the acoustic properties, such as relative frequency response [43], [44], echo traces [45], and trawl sampling [46]. For that reason, the manual process is vulnerable to bias from the operators. In extreme cases, the systematic error associated with the manual method can be as high as ± 80% [8]. Hence, more structured and automated approaches are required to apply consistent criteria to the analysis while reducing dependence on human intervention. To this end, postprocessing systems, including the large scale survey system (LSSS) [9], are developed to facilitate the manual process. The systems support thresholding, error-checking, noise removal, and manipulation of the echosounder data. By adjusting the threshold of backscattered intensities, the postprocessing systems visualize the corresponding morphology of the fish schools to enable the operators to detect and delineate the most plausible morphology. In addition, these postprocessing systems enable relatively consistent criteria for the analysis by leveraging their acoustic feature libraries. The library consists of a selected part of the backscattered responses and their manually annotated class attributes. By comparing the statistical properties of the collected data to the feature library, the postprocessing system predicts the class attribute of the fish school, where the prediction is verified by the scattering model for the corresponding marine organism if available [47], [48].

The sandeel data in this study are manually annotated with the aid of LSSS, where expert operators determine the class of each backscattering coefficient as sandeel (SE), other fish species (OT), or background (BG) class. The primary frequency for LSSS is chosen to 200 kHz considering the highest sandeel signal-to-noise ratio [49]. The operators alter the detection threshold centered at -63 dB at the primary frequency to discover the fish school boundaries visually. The delineated boundary is refined using binary morphological closing to have smoother and pragmatic edges [5]. However, the final decision for both morphology and species is still a manual process, which is time-consuming and requires tacit knowledge that can be potentially biased as with any expert system.

Therefore, recent studies have focused on the automated identification of the fish species using machine-learning methods while leveraging the conventional detection algorithm to detect and delineate the morphology of the schools. Shoal analysis and patch estimation system (SHAPES) [50], [51] is often chosen for the fish school detection algorithm, which extracts a feature vector from each fish school leveraging a single frequency channel of 38 kHz. A random forest-based classifier [12] is introduced to classify feature vectors of silver cyprinid from the other species in Lake Victoria. Aronica et al. [52] propose a classifier leveraging a shallow feedforward network and classify the pelagic Mediterranean fish schools such as anchovy, sardine, and horse mackerel. Those studies show that the automated identification can save time and cost while also achieving robust performance. However, they have limitations in generalizability and scalability because the SHAPES algorithm only exploits a single channel of the echosounder data, and a handcrafted feature selection is required to improve the performance.

Deep learning-based models generalize and scale well on various types of data using their flexibility [15], [37]. Among them, the fully supervised deep learning approaches, approaches that learn from the fully annotated training data, achieve a good level of performance provided a high quality of the training data and an appropriate choice of the prediction model are assured. To take advantage of supervised deep learning in the analysis of echosounder data, CNN-based semantic segmentation model [5] is introduced to segment the schools of lesser sandeel from the other species leveraging the U-Net architecture [23]. Without relying on the deterministic school detection algorithms and the feature vectors as input, the model constructs the prediction map directly from the input echosounder data.

B. Deep Clustering

We here discuss *deep clustering* since our novel CNN-based semisupervised semantic segmentation for echosounder data, presented in Section III, relies heavily on this concept. Deep clustering refers to unsupervised deep learning-based approaches, that aim to cluster data into underlying groups without requiring the class attributes of the data [53]. Deep clustering leverages the representation power of the neural network in conjunction with clustering algorithms, and partitions the input data into clusters with respect to the learned representation. As clustering performance heavily depends on the underlying structure of the data, deep clustering leverages the neural network to encode the training images in the feature representations where the clustering task becomes much easier [54].

Our proposed method is inspired by a well-known deep clustering framework, referred to as DeepCluster [53], which explicitly models the density of datapoints leveraging the k-means clustering algorithm. For a given image data set, the k-means algorithm partitions the feature representation into K different densities, where each density refers to an image descriptor or a visual feature. This has the advantage that it is easy to increase the capacity of more visual features by simply increasing the number of clusters K, leading to all-purpose visual features. The neural network produces cluster indices that can be thought of as clustering-induced annotations for the training data. The network is then updated in a supervised manner to learn the

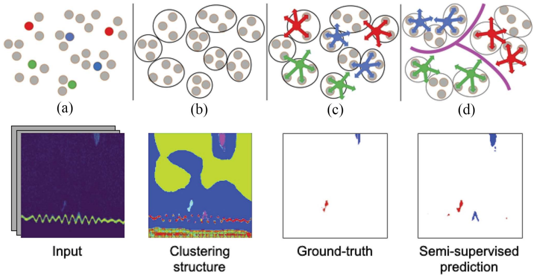


Fig. 1. Overview of the proposed method. Each backscattering intensity in the input is mapped into an arbitrary space shown in (a). The point in gray is unannotated while the point in color (red, green, or blue) indicates the annotated one with respect to the class. (b) Clustering structure incorporated by the unsupervised clustering objective without leveraging the annotation. The clustering structure becomes the pseudolabel to train the model in an unsupervised manner. (c) indicates that the annotated data (ground-truth where available) and the supervised segmentation objective optimize the CNN in a supervised manner. (d) indicates that the iteration of (b) and (c) constructs the decision boundary with respect to given classes, where the unannotated points take their place inside the boundary according to their own clusters.

clustering structure. This annotation technique is referred to as pseudolabeling, allowing the supervised deep learning approach to be applied to unannotated training data [55].

III. Proposed Method

In this article, we propose a novel semisupervised semantic segmentation method, PredKlus, that enables a CNN to simultaneously learn from large amounts of unannotated data and a few annotated data, all in the same network.

The major novelty of our work is the methodology of how the network learns in a semisupervised manner, illustrated in Fig. 1. Our proposed segmentation network operates for two different goals: 1) searching for the internal structure of the training data without relying on external information, e.g., ground truth; 2) mapping input echosounder data to given classes. The former goal can be achieved by an *unsupervised clustering objective*, which clusters every pixels in the input based on their features to reveal a clustering structure of the input data in an unsupervised manner. Fig. 1(b) illustrates the clustering structure. A *supervised segmentation objective*, on the other hand, aims to map the input to given classes by leveraging the annotated part of training data, albeit in a small amount. Fig. 1(c) illustrates this. As these two objective functions alternately optimize the network using gradient descent, the segmentation network gradually learns the class decision boundaries (supervised) with respect to the clustering structure (unsupervised), as illustrated in Fig. 1(d). We implement the entire learning process in an end-to-end manner and a mini-batch setting, which are additional novelties of our method.

A. Model Architecture

Fig. 2 describes the model architecture of our proposed method. The encoder–decoder architecture with the skip connections is inspired by U-Net [23] and the recent segmentation study of the echosounder data [5]. The encoder part extracts the abstracted feature map of the echosounder input with a shape of $256 \times 256 \times 4$ over five stages, where the area of the feature map is reduced to one-fourth at each stage due to a 2×2 max-pooling layer. By processing two sets of a 3×3 convolutional layer, a batch-normalization layer [56], and a rectified linear unit (ReLU) [57] at each stage, we abstract the feature map by doubling the depth. The encoder eventually creates five feature maps of different area sizes and depths, where the shape of the last feature map is $16 \times 16 \times 1024$.

The decoder part reconstructs the prediction map leveraging five feature maps from the encoder. At each stage, a 2×2 transposed convolutional layer and the concatenation of the feature maps along the depth axis play an important role. The 2×2 transposed convolutional layer increases the area of the feature map fourfold while halving the depth. The halved feature map is concatenated with the feature map in the same shape from the encoder. The concatenated feature map is processed by two sets of a 3×3 convolutional layer, a batch-normalization layer, and an ReLU, where the depth becomes halved.

The novelty in our architecture is to introduce a convolutional layer for each objective function at the end of the CNN to employ two objective functions in one network. The alternation of the two objective functions takes place at the end of the decoder, where the decoder reconstructs the feature map with a shape of $256 \times 256 \times 64$. To alternately leverage two objective functions, we append a 1×1 convolutional layer at the end of the network for each objective function, namely, *conv1* for the unsupervised clustering objective and *conv2* for the supervised segmentation objective. Note that the number of filters in *conv1* matches the number of clusters or pseudoclasses K. Similarly, the number of filters in *conv2* is equal to the number of classes C.

B. Two Objective Functions

Our proposed method leverages two objective functions, where those objectives alternately optimize the model. Through the alternating optimization, the CNN indirectly incorporates the class annotations (supervised) to a structured representation (unsupervised) and eventually discovers a structured representation consistent with the available annotations. The yellow box in the middle of Fig. 2 shows the overview of our semisupervised segmentation method. The first two steps of the figure, i.e., Fig. 2(a) creating pseudolabels using k-means, and Fig. 2(b) updating the model to learn the clustering structure with the pseudolabels using *conv1*, contribute to learning the structured representation in an unsupervised manner. The next step, Fig. 2(c) training with the partially available annotation using *conv2*, represents how the CNN learns in a supervised manner using the supervised segmentation objective and the available class annotations. Note that a cross-entropy loss (CE) is leveraged to update the model, as depicted in Fig. 2(b) and (c).

1) Unsupervised Clustering Objective: The unsupervised clustering objective exploits the underlying structure of the data using the unsupervised clustering algorithm, such as k-means, to create pseudolabels with respect to the clustering structure [53]. Defining the number of clusters K beforehand, the proposed

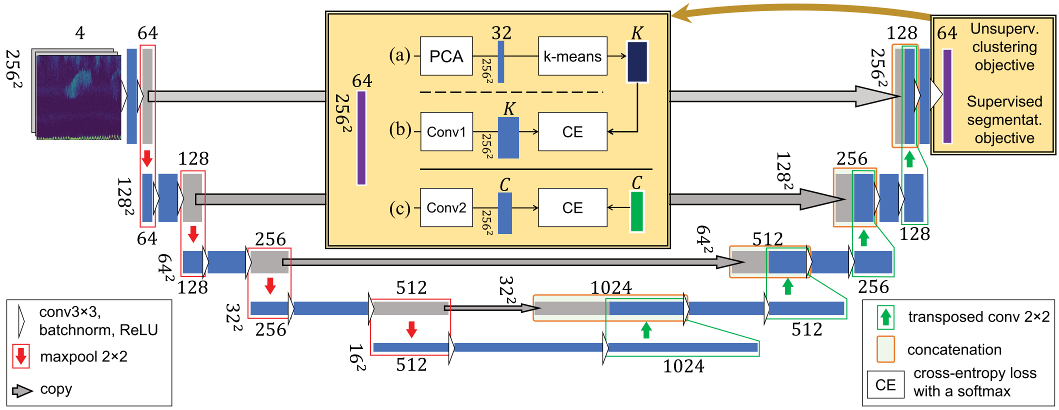


Fig. 2. Proposed model architecture. The application of the two objective functions takes place at the yellow box at the end of the decoder. The unsupervised clustering objective involves in the first two steps. (a) Creating pseudolabel using k-means and (b) updating model to learn the clustering structure with the pseudolabel using *conv1*. The supervised segmentation objective involves in (c) training with the partially available annotation using *conv2*. The rectangular bars in blue or gray represent feature maps, where the size of each feature map is specified around it, e.g., 256^2 or 16^2. We omit to specify the depth for a few feature maps, as the depth is the same as the feature map on its right, e.g., 64 or 512.

model partitions the feature map $\mathcal{Z} = \{\mathbf{z}^{(i)}\}_{i=1}^{N}$ located at the end of the decoder into K clusters in a way to find the best assignment by minimizing the k-means loss

$$\mathcal{L}_{kmns} = \frac{1}{N}\sum_{i=1}^{N} \min_{\mathbf{c}_k} d\left(\mathbf{z}_{PC}^{(i)}, \mathbf{c}_k\right). \tag{1}$$

In this expression, N is the number of feature vectors in a mini-batch of the feature map. If the batch size B_s is equal to one, N becomes 65 536 as each feature map consists of 65 536 vectors (256 × 256). The function $d(\cdot,\cdot)$ measures the L_2 distance between two vectors, where $\mathbf{c}_k \in \mathbb{R}^{32}$ is the centroid of cluster k, and $\mathbf{z}_{PC}^{(i)} \in \mathbb{R}^{32}$ is the dimensionality-reduced training set consisting of the feature vectors $\mathbf{z}^{(i)} \in \mathbb{R}^{64}$. For dimensionality reduction, we use principal component analysis (PCA) [58], which computes the principal components and use only the first few principal components corresponding to the largest eigenvalues for manageable computational complexity.

The clustering result creates the pseudolabels, having K different pseudoclass attributes according to the K cluster indices. The CNN learns the structured representation from the pseudolabels using the cross-entropy loss. The unsupervised clustering objective is depicted as

$$\mathcal{L}_{cls} = \frac{1}{N}\sum_{i=1}^{N} w_{cls,k}^{(i)} CE\left\{g_\theta(\mathbf{z}^{(i)}), \hat{\mathbf{y}}^{(i)}\right\} \tag{2}$$

where $CE(p,q) = -\sum_k q_k \log(p_k)$ is the cross-entropy loss of the probability distribution p for the one-hot encoded label q, $\hat{\mathbf{y}}^{(i)} \in \{0,1\}^K$ is the one-hot encoded pseudolabel, and $g_\theta(\mathbf{z}^{(i)})$ is a probability distribution of the output from the CNN, where *conv1* is appended at the end of the decoder. The scalar $w_{cls,k}^{(i)}$ indicates the class-rebalancing weight to penalize the class imbalance of the pseudolabels. How to obtain this scalar will be explained in Section III-C. Once updating the CNN with the unsupervised clustering objective, we assign the current centroids of K clusters to the initial centroids for the next clustering to provide consistency of the pseudolabels over the mini-batches.

2) Supervised Segmentation Objective: To enforce consistency of predictions with regard to the given classes, we train the CNN using the partially available annotated data. The supervised segmentation objective is involved here, where *conv2* layer, another 1 × 1 convolutional layer, replaces the *conv1* layer to allow end-to-end training. The supervised segmentation objective is depicted as

$$\mathcal{L}_{seg} = \frac{1}{N}\sum_{i=1}^{N} w_{seg,c}^{(i)} CE\left\{f_\theta(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}\right\} \tag{3}$$

where C represents the number of given classes, $\mathbf{y}^{(i)} \in \{0,1\}^C$ represents the one-hot encoded vector of the available annotation. $f_\theta(\mathbf{x}^{(i)})$ a probability distribution of the output from the CNN, where *conv2* replaces *conv1*.

3) Training Procedure: In addition to end-to-end learning, the proposed method operates in a mini-batch training manner, indicating that the network is updated once after each objective processes information in each mini-batch [59]. We form two training subsets for each objective function to facilitate the alternating mini-batch training. The training subset for the unsupervised clustering objective consists of the entire training input data, whether annotated or not, and does not include any class annotation of the data. On the other hand, the training subset for the supervised segmentation objective includes the annotated part of the training data, which takes a small amount of the entire training data in the semisupervised learning scheme. Algorithm 1 illustrates the semisupervised training procedure with two training subsets.

Algorithm 1: Training by Alternating Two Objectives.

```
Input:
$\mathcal{X}$: training input data
$\mathcal{X}^A \subset \mathcal{X}$: the annotated part of the training input data
$\mathcal{Y}^A$: class annotation of $\mathcal{X}^A$
$\mathbf{X}$: an unannotated mini-batch of $\mathcal{X}$
$(\mathbf{X}^A, \mathbf{Y}^A)$: an annotated mini-batch of $\mathcal{X}^A$ and $\mathcal{Y}^A$
Output:
$\mathbf{Z}$: feature map of the mini-batch $\mathbf{X}$ at the end of the decoder
$\hat{\mathbf{Y}}$: created pseudolabel of the mini-batch $\mathbf{X}$
$\mathbf{P}^A$: class prediction of the mini-batch $\mathbf{X}^A$
Procedure:
  for $(\mathbf{X}, \mathbf{X}^A, \mathbf{Y}^A) \in (\mathcal{X}, \mathcal{X}^A, \mathcal{Y}^A)$ do
    – Compute $\mathbf{Z}$ by processing $\mathbf{X}$ through the model
    – Create pseudolabel $\hat{\mathbf{Y}}$ by clustering the principal components of $\mathbf{Z}$
    – Compute $w_{cls}$ with respect to $\hat{\mathbf{Y}}$
    – Append conv1 at the end of the decoder
    – Update the model end-to-end using $(\mathbf{X}, \hat{\mathbf{Y}})$ and the unsupervised clustering objective in (2) with gradient descent
    – Replace conv1 by conv2
    – Compute $\mathbf{P}^A$ by processing $\mathbf{X}^A$ through the model
    – Compute $w_{seg}$ with respect to $\mathbf{P}^A$
    – Update the model end-to-end using $(\mathbf{X}^A, \mathbf{Y}^A)$ and the supervised segmentation objective in (3) with gradient descent
  end for
```

C. Advance on the Semisupervised Image Classification for Echosounder Data [22]

The problem of being able to obtain manual annotations is much more severe for semantic segmentation compared to image classification, since in the former case annotations refer to the pixel level and not the entire image. The semisupervised method we propose in this article therefore solves a much more challenging problem compared to our previous preliminary work on semisupervised echosounder data patch classification [22], which is only able to classify whole image patches and not do proper segmentation. Some elements of the new segmentation method resembles the previous classification method, however with significant differences due to the completely different aims of the two approaches. For the benefit of the reader, and since we use [22] as one of the comparison models in experiments (referred to as SemiClf, Section IV-D), we will elaborate on these differences in this section.

SemiClf [22] is an image classification method, which is also semisupervised by design, built around two alternating objective functions. However, this semisupervised algorithm has some critical drawbacks. The minimum patch size that the method can classify is 32 × 32 intensity pixels. This is far too coarse-grained to provide information at a pixel level. Second, the training procedure is inefficient. During training, the method samples the patches to tackle the imbalance in the cluster size. The sampling hinders mini-batch training, degenerating training efficiency. We highlight benefits of our new semantic segmentation method below.

TABLE I
Overview of the Echosounder Data Used for Training and Test/Validation

Year	Training set (2016–2017)	Test set (2019)
No.patches	200	60
The number of backscattering intensities per class (proportion)		
BG	12 995 258 (0.9914)	3 904 023 (0.9928)
SE	61 018 (0.0047)	11 776 (0.0030)
OT	50 924 (0.0039)	16 361 (0.0042)
Total	13 107 200 (1.0000)	3 932 160 (1.0000)

1) Obtaining Fine-Grained Segmentation Maps: SemiClf [22] classifies echosounder patches with a shape of 32 × 32 × 4 into three classes using the modified architecture of VGG-16 [60], where 4 in the patch shape indicates the number of frequency channels. The architecture corresponds to an encoder of the neural networks. The result can be interpreted as a coarse-grained segmentation, where the minimum resolution of prediction is equal to the patch shape. On the contrary, our method leverages the modified U-Net architecture [23], providing a fine-grained segmentation where the minimum resolution is 1 × 1 × 4.

Training the CNN for semantic segmentation is much more challenging than the one for classification because the large and sophisticated architecture may hinder the backpropagation of the gradient to the other end of the network. We leverage the coupled architecture of encoder and decoder using dilations and concatenation functions to facilitate the backpropagation of the gradient, as suggested in U-Net. In addition, we simplify the data preprocessing by avoiding applying the criteria for determining which class each patch belongs to, which is required for the classification task.

2) Annotation-Free Class-Rebalancing Weight: Our method utilizes the cross-entropy loss for both the unsupervised and supervised learning schemes. However, the cross-entropy loss does not account well for imbalanced classes as it sums over all the intensities [61]. A common approach to tackle the class imbalance problem is to allocate class importance to mitigate the imbalance based on the class distribution. This includes rebalancing the class weights [62], [63], [64] and regulating the learning frequency by sampling [22], [53], [65]. Table I shows that the echosounder data are severely class-imbalanced to the given classes, where more than 99% of the backscattering intensities belong to the background (BG) class consisting of the water and seabed features. The supervised segmentation objective, therefore, should deal with the class imbalance problem in the echosounder data.

The unsupervised clustering objective should also tackle the class imbalance problem. The clustering approaches based on DeepCluster [53] can result in a trivial solution, such as empty clusters or immensely larger clusters than their average size. This causes the imbalance among the pseudoclasses, hindering the CNN to address the structured representation. To tackle the imbalance, approaches based on DeepCluster [22], [53] purposely equalize the cluster size by sampling to uniformly distribute the pseudoclasses. For the segmentation task, however, sampling pixels to create the class-balanced pseudolabels is not a strategic choice in terms of the learning efficiency as the discarded pixels may create a mask in the pseudolabel, hindering end-to-end mini-batch training.

Hence, we apply the class-rebalancing weight technique to the objective functions to bypass the sampling procedure. The weight leverages the number of predictions to each pseudoclass or class attribute instead of leveraging the available class annotation, differentiating our method from the previous studies [5], [22]. The class-rebalancing weight $w_{cls,k}$ for the unsupervised clustering objective $\mathcal{L}_{cls}$ in (2) is depicted as

$$w_{cls,k} = \frac{\hat{w}_{cls,k}}{\sum_{k \in K} \hat{w}_{cls,k}}, \text{ where } \hat{w}_{cls,k} = \frac{N}{KN_k}. \tag{4}$$

In this expression, N represents the total number of pseudolabels in a mini-batch. K represents the number of pseudoclasses or clusters that we predefined. N_k represents the number of pseudolabels of the pseudoclass k, where the sum over the K pseudoclasses is equal to N $(N = \sum_{k \in K} N_k)$. Equation (4) indicates that the pseudoclasses larger than the average size N/K are penalized by the smaller weight than the other classes.

In this study, rather than forcing the balance in a few available annotations, we introduce the class-rebalancing weight $w_{seg,c}$ for the supervised segmentation objective $\mathcal{L}_{seg}$ in (3) depicted as

$$w_{seg,c} = \frac{\hat{w}_{seg,c}}{\sum_{c \in C} \hat{w}_{seg,c}}, \text{ where } \hat{w}_{seg,c} = \frac{N}{CN_c}. \tag{5}$$

In this expression, C represents the number of classes in the annotated data. N_c represents the number of prediction of the class c, where the sum over the C classes is equal to N $(N = \sum_{c \in C} N_c)$. Note that we count N_c from the prediction of the model rather than the available annotation to avoid the deterministic weight values, resulting in the annotation-free class rebalancing weight.

IV. Experiment

The purpose of the experiments is to explore the robustness of the proposed method in the semisupervised learning environment that exploits limited annotations and, at the same time, the contribution of the unannotated data. We evaluate our method by comparing it with other segmentation models applied for the analysis of the echosounder data, where the evaluation metrics include prediction accuracy, F1-score, confusion matrix, Cohen's kappa [66], and area under the curve–receiver operating characteristics (AUC-ROC) [67].

A. Data Setup

We leverage the echosounder data from 2016 to 2017 to train the CNN-based segmentation model and the trained model is evaluated using the echosounder data from 2019. The size of the input echosounder patches is 256 × 256 × 4, where 4 indicates the number of echosounder channels (18, 38, 120, 200 kHz). We randomly extract the echosounder patches from the echosounder data. 200 patches from the echosounder data between 2016 and −2017 are used for the training set, and 60 patches from the echosounder data in 2019 are used for the test set. In addition to those sets, we extract 30 patches for the validation set from the echosounder data between 2016 and 2017 to tune the hyperparameters. There is no overlap among the patches. The model output is the segmentation map of the corresponding input, segmented by the three given classes. Table I and Fig. 3 show, respectively, a subset of the training patches and the general information of the training and test sets.

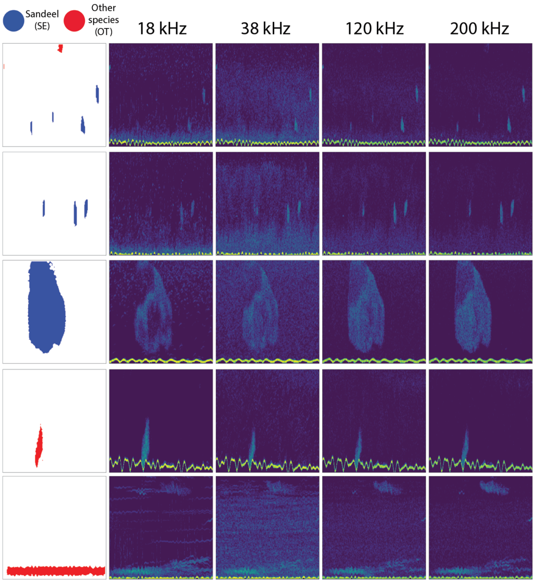


Fig. 3. Five pairs of the training patches. The annotation map (leftmost) and the echosounder data for each frequency channel are shown. The colors in the annotation map indicate the classes: background (BG) in white, other fish species (OT) in red, and sandeel (SE) in blue. The horizontal yellow line at the lower part of the echosounder data is the boundary between water and the seabed. Note that some patches do not include any fish pixel as a result of random patch extraction.

B. Annotation Ratio

To explore the impact of our semisupervised method, we compute the annotation ratio, which measures the ratio of the number of annotated patches to the number of the entire set of training patches. Six ratios are studied, namely, 1.00, 0.40, 0.35, 0.30, 0.25, and 0.20. The annotation ratio of 1.00 represents a fully supervised setting, where 200 training patches are fully annotated. The annotation ratio of 0.20 takes the extreme semisupervised case in this study, where 40 out of 200 training patches are annotated while the remaining 160 patches are unannotated.

TABLE II
PERFORMANCE COMPARISON REGARDING DIFFERENT KS AT THE ANNOTATION RATIO OF 0.20

0.20	No.clusters (K)	256	512	1024	2048
AUC-ROC	BG	0.7582	0.8672	**0.9407**	0.9258
	SE	0.7033	**0.8306**	0.7075	0.7585
	OT	0.7873	0.7851	**0.8559**	0.6523
Accuracy	BG	0.9850	**0.9861**	0.9809	0.9847
	SE	0.4628	**0.5312**	0.4731	0.5166
	OT	0.4657	0.5224	**0.5817**	0.4881
Kappa		0.2991	**0.3449**	0.3045	0.3374
F1		0.9844	**0.9856**	0.9828	0.9849

The bold values denote statistical significance at the p < 0.05 level.

C. Training Configuration

The following training configuration is shared for all experiment setups. The model learns by mini-batch training, where the batch size B_s is set to 2 considering the computational resource. Thus the number of feature representations in a mini-batch N is 131 072 (2 × 256 × 256). The Adam optimizer [68] with learning rate 3 × 10^{-5}, beta (0.9, 0.999), and weight decay 10^{-5} is applied. The training is iterated to 500 epochs for all experiments, applying early stopping [69] on the condition that the accuracy is not improved for 100 epochs. For PCA, we choose the first 32 principal components shown in (1) as they capture most of the variance of the data. Three prediction classes are given ($C = 3$); background (BG), sandeel (SE), and other fish species (OT).

Regarding the choice of the number of clusters K, we choose $K = 512$ after testing a set of different Ks. Table II exemplifies one of the tests when the annotation ratio is 0.20, where the AUC-ROC value of SE class (0.8306), prediction accuracy of BG and SE classes (BG accuracy 0.9861; SE accuracy 0.5312), Cohen's kappa (0.3449), and F1 score (0.9856) achieve the highest when $K = 512$. As addressed in the DeepCluster work [53], the number of cluster K does not have a significant impact on the performance if we cluster the feature representations with a sufficiently large number of clusters compared to the number of classes. We tune those hyperparameters using the validation set. All the code is implemented in PyTorch [70].

D. Validation Methods

Our proposed method, PredKlus, is designed specifically to exploit the intrinsic nature of unannotated data, as well as to enforce class structure by supervision, all while handling the inherent class-imbalance of echosounder data by class-rebalancing weights. One could envision other approaches for exploiting unannotated data in semantic segmentation for acoustic target detection.

As the first comparison model to highlight this, we reimplement a recently published work for generic semisupervised semantic segmentation [71] for our specific task of acoustic target classification. This method, which we refer to as SemiCPS, also aims to integrate pseudoclass predictions (unsupervised) to the class predictions (supervised) by introducing an additional auxiliary segmentation network mirroring the main segmentation network architecture with different initializations.

SemiCPS intends to encourage high similarity between the predictions of the two networks with different initialization for the same input image. For the annotated input, each network is individually trained in a supervised manner. For the unannotated input, the main network first creates the class prediction map by processing the input. This prediction map becomes the pseudolabel that will supervise the auxiliary network. Once the auxiliary network is updated by the pseudolabel, the main network is also supervised by the prediction map from the auxiliary network.

With SemiCPS, we implicitly explore how the unsupervised clustering objective affects the predictive performance when data are noisy. Due to the unpredictable underwater nature, the features between the target class and the nontarget are visually indistinguishable in some echosounder data. This may lead the mirrored network of SemiCPS to generate incorrect pseudolabels, which are tied to the supervision of the main network. If it eventually repeats, none of the two networks can make correct predictions. On the other hand, the pseudolabels in our proposed method are leveraging the internal structure of the data set and are not tied to the class supervision. This makes our proposed approach more robust against noisy data, such as the echosounder data. As we will show in Section V, SemiCPS does not compare favorably to our approach. We believe this to be due to an inability to exploit the intrinsic nature of the unannotated data leading to a propagation of errors induced by the pseudolabeling due to the noisy nature of the data. This will be further discussed in Section V-A.

The second comparison model is the semisupervised patch classification method [22], referred to as SemiClf, where both the annotated and the unannotated parts are involved in the analysis. This model learns from a small input patch of size 32 × 32 × 4, and classifies each patch to given classes leveraging the architecture of the modified VGG-16 [60]. We train SemiClf using the same training set, after splitting one provided echosounder input (256 × 256 × 4) into 64 small patches. In the inference phase, on the other hand, we extract the small patches with stride of one pixel only, resulting in a fine-grained prediction map. A voting mechanism determines the class for each pixel, which is based on the class prediction frequency among the overlapping small patches. This significantly increases the computational complexity of SemiClf, but provides a pixel-level comparison between all methods.

The third comparison model is the fully supervised segmentation method [5], referred to as SupSeg in this study. This utilizes the same CNN architecture and the supervised segmentation objective as our proposed method, and provides the class prediction of each backscattering intensity. But it does not exploit either the unannotated part of the data or the unsupervised clustering objective. For semisupervised settings where the annotation ratios are smaller than one, this fully supervised method ignores the unannotated part and learns from the annotated part of the training set, which is partially available.

TABLE III
MODEL PERFORMANCE COMPARISON WITH RESPECT TO AUC-ROC VALUE AND CLASS ACCURACY

Anno.	Class	AUC-ROC				Accuracy			
ratio		Ours	SupSeg	SemiClf	SemiCPS	Ours	SupSeg	SemiClf	SemiCPS
0.20	BG	**0.8672**	0.8331	0.7870	0.6369	**0.9861**	0.9833	0.7105	0.8472
	SE	0.8306	0.6576	**0.8496**	0.6229	0.5312	0.4813	**0.6867**	0.3253
	OT	**0.7851**	0.6816	0.4668	0.1991	**0.5224**	0.4906	0.0146	0.0000
0.25	BG	**0.8499**	0.8457	0.8390	0.6510	**0.9880**	0.9877	0.7748	0.9851
	SE	**0.7952**	0.7251	0.7606	0.5792	**0.5290**	0.5120	0.1208	0.1416
	OT	0.7879	0.7762	**0.8387**	0.3886	0.5340	0.5271	**0.6726**	0.0000
0.30	BG	**0.9148**	0.8763	0.9019	0.7468	**0.9856**	0.9851	0.8530	0.9155
	SE	**0.8387**	0.8052	0.8240	0.6005	**0.6282**	0.6080	0.6639	0.3295
	OT	**0.8423**	0.7744	0.7792	0.6834	**0.5326**	0.5231	0.0501	0.2115
0.35	BG	**0.9385**	0.8474	0.8666	0.7945	0.9842	**0.9857**	0.7444	0.8859
	SE	**0.8687**	0.7977	0.7770	0.8159	**0.6609**	0.6128	0.5938	0.5670
	OT	**0.8930**	0.8856	0.8103	0.5836	**0.6419**	0.6399	0.1329	0.1792
0.40	BG	**0.9097**	0.9015	0.8446	0.8455	0.9811	**0.9857**	0.9534	0.8846
	SE	**0.8840**	0.8103	0.8256	0.8539	**0.6304**	0.6238	0.3769	0.5671
	OT	**0.8621**	0.8128	0.7968	0.7572	**0.7307**	0.6029	0.1748	0.3226
1.00	BG	**0.9262**	0.8696	0.8651	0.9088	**0.9888**	0.9886	0.8602	0.8687
	SE	**0.8705**	0.8619	0.8221	0.8634	**0.6779**	0.6076	0.3420	0.5247
	OT	0.9025	0.8285	0.8135	**0.9045**	**0.7461**	0.7180	0.4489	0.6548

The bold values denote statistical significance at the $p < 0.05$ level.

TABLE IV
MODEL PERFORMANCE COMPARISON WITH RESPECT TO COHEN-KAPPA AND F1 SCORE

Anno.	Cohen-kappa				F1 score			
ratio	Ours	SupSeg	SemiClf	SemiCPS	Ours	SupSeg	SemiClf	SemiCPS
0.20	**0.3449**	0.3267	0.0191	0.0183	**0.9856**	0.9843	0.8208	0.9047
0.25	**0.3756**	0.3747	0.0383	0.0454	**0.9869**	0.9868	0.8634	0.9778
0.30	**0.3579**	0.3558	0.0571	0.0560	**0.9857**	0.9854	0.9107	0.9436
0.35	0.3774	**0.3887**	0.0306	0.0515	0.9855	**0.9862**	0.8448	0.9276
0.40	0.3565	**0.3878**	0.0951	0.0664	0.9841	**0.9863**	0.9638	0.9277
1.00	**0.4796**	0.4540	0.0596	0.0770	**0.9889**	0.9885	0.9143	0.9194

The bold values denote statistical significance at the $p < 0.05$ level.

V. RESULT AND DISCUSSION

Our method and three comparison models, e.g., SupSeg [5], SemiClf [22], and SemiCPS [71], are evaluated by the various performance measures using the test echosounder data specified in Table I. The measures include AUC-ROC value and the class prediction accuracy for each class and annotation ratio (Table III), Cohen's kappa (kappa), and F1 score regarding each annotation ratio (see Table IV). The area under the ROC curve is AUC, where a higher AUC indicates better segmentation performance. Regarding the class prediction accuracy, note that the SE class achieves the lowest prediction accuracy than any other class for the many setups. This indicates that the SE class is a conservative estimate [22].

In addition to these measures, the confusion matrix and the corresponding ROC curve for each setup are computed for the comparison, as shown in Figs. 4–9. For the confusion matrices, each row of these confusion matrices sums to one, indicating the ground truth of the prediction. Each column illustrates the class prediction of the method. The first column and row indicate the BG class, the second and the third columns and rows denoting the SE class and the OT class, respectively. For the ROC curves, the vertical axis indicates a true-positive rate while the horizontal axis shows a false-positive rate. For the visual comparison, we provide the prediction map of the test data in Figs. 10–12, where four parts of the echosounder data in 2019 and their prediction maps are visualized. Overall, the results show that our semisupervised method outperforms the comparison models throughout annotation ratios.

A. *Comparison to Semisupervised Segmentation Method Using Pseudolabels (SemiCPS)*

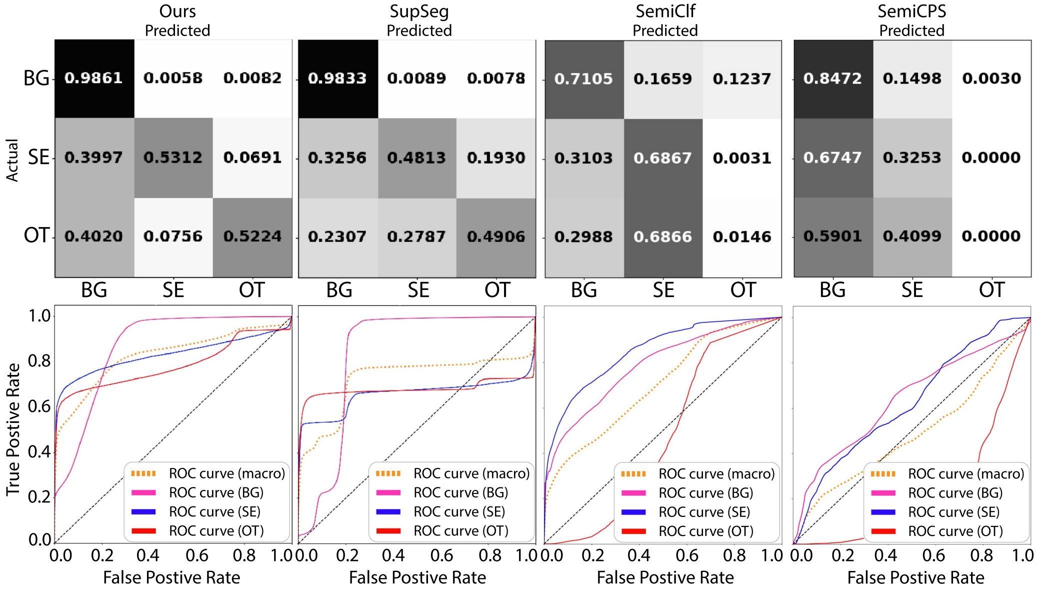
Tables III and IV show that our proposed method outperforms SemiCPS through the entire evaluation metrics in the semisupervised setups containing the annotation ratios of 0.20–0.40. The greatest performance difference is observed at the annotation ratio of 0.20, which is the most extreme semisupervised setup. Our method achieves the kappa score of 0.3449, which is 18.8 times greater the kappa score of SemiCPS (0.0183).

The prediction maps in Fig. 10 also visually validate the outperforming results of our proposed method. SemiCPS does not make predictions close to the fish patterns for the annotation ratios of 0.20–0.25, but tends to capture the fish class patterns from the annotation ratios of 0.30 and higher. However, quite a few fish patterns are still misclassified to the BG class, yielding a smaller prediction area and underperforming results than our proposed method. Our proposed method, in contrast, tends to capture most of the major fish patterns on the prediction map from the annotation ratio of 0.20. Although the prediction map appears noisy due to misclassification of small clutter patterns at low annotation ratios, the noise is filtered out as the annotation ratio increases and shows a good prediction map close to the ground truth and the input. We discover the same visual trends in Figs. 11 and 12.

B. *Comparison to Semisupervised Patch-Based Segmentation (SemiClf)*

Compared to SemiClf [22], our proposed method outperforms throughout the measures and setups. We argue that the novelties of our method, such as the learning mechanism for the fine-grained segmentation and the annotation-free class-rebalancing technique, contribute to achieving the surpassing result by addressing the shortcomings of patch-based SemiClf. The kappa scores contrast the difference nicely, where ours achieves 18.3 times greater scores than SemiClf with the annotation ratio of 0.20 (ours 0.3449; SemiClf 0.0191).

In addition to the poor prediction maps shown in Figs. 10–12, another critical drawback of SemiClf is misclassification of the seabed feature, which is known for a considerably higher intensity than the other fish classes [5]. The seabed feature is marked with a distinct yellow horizontal line in the input echosounder data. As shown in the prediction maps, SemiClf and SemiCPS predict the seabed as one of the fish classes (blue or red) throughout the annotation ratios. In contrast, our method learns the seabed feature and correctly predicts it to BG class in white as intended.



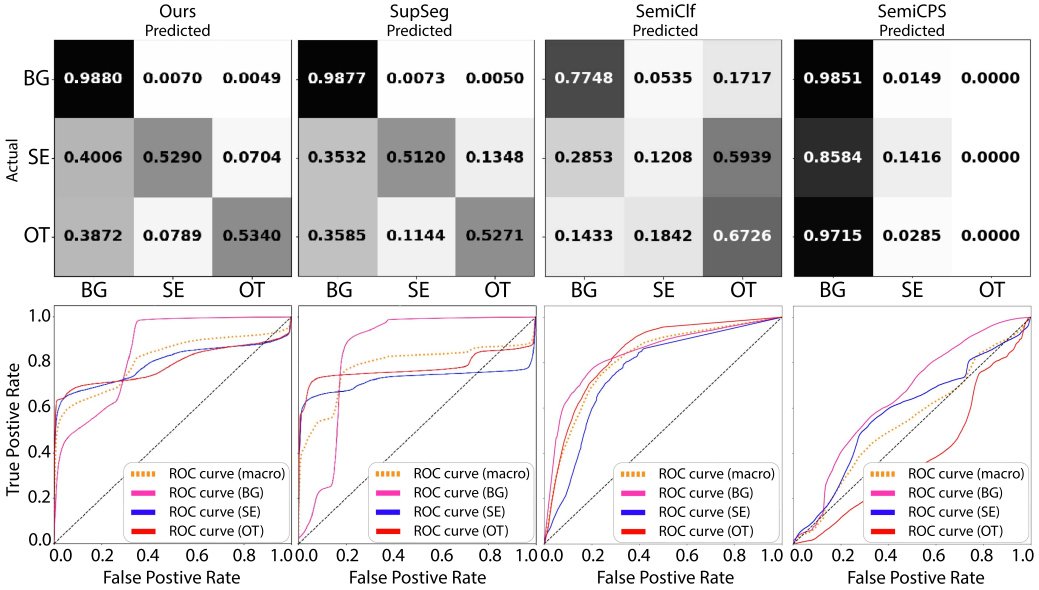
Ours	BG	SE	OT
BG	0.9861	0.0058	0.0082
SE	0.3997	0.5312	0.0691
OT	0.4020	0.0756	0.5224

SupSeg	BG	SE	OT
BG	0.9833	0.0089	0.0078
SE	0.3256	0.4813	0.1930
OT	0.2307	0.2787	0.4906

SemiClf	BG	SE	OT
BG	0.7105	0.1659	0.1237
SE	0.3103	0.6867	0.0031
OT	0.2988	0.6866	0.0146

SemiCPS	BG	SE	OT
BG	0.8472	0.1498	0.0030
SE	0.6747	0.3253	0.0000
OT	0.5901	0.4099	0.0000

Fig. 4. Confusion matrices and the corresponding AUC-ROC plots of the annotation ratio of 0.20.



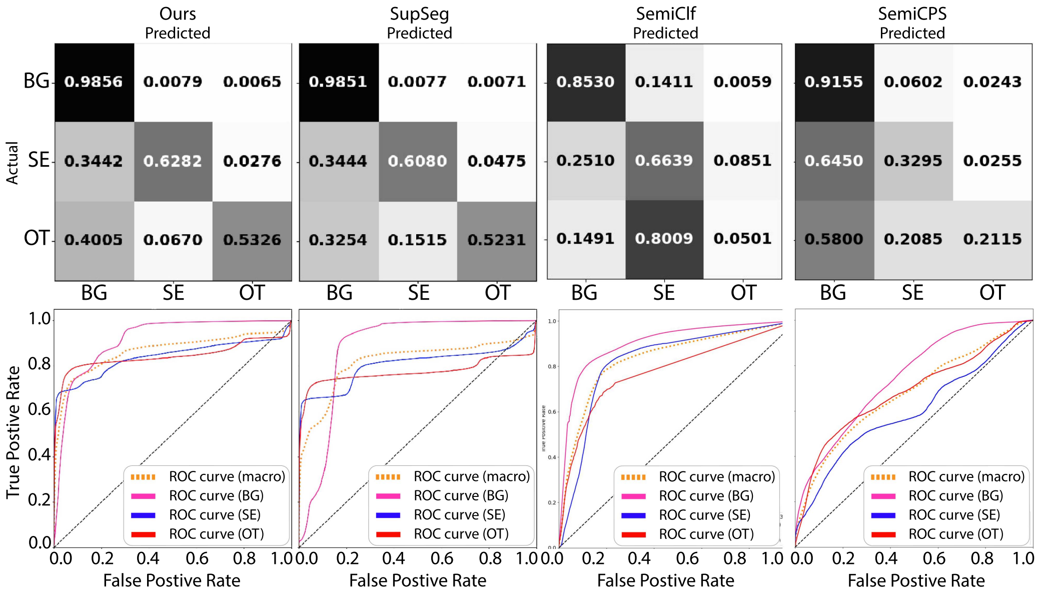
Ours	BG	SE	OT
BG	0.9880	0.0070	0.0049
SE	0.4006	0.5290	0.0704
OT	0.3872	0.0789	0.5340

SupSeg	BG	SE	OT
BG	0.9877	0.0073	0.0050
SE	0.3532	0.5120	0.1348
OT	0.3585	0.1144	0.5271

SemiClf	BG	SE	OT
BG	0.7748	0.0535	0.1717
SE	0.2853	0.1208	0.5939
OT	0.1433	0.1842	0.6726

SemiCPS	BG	SE	OT
BG	0.9851	0.0149	0.0000
SE	0.8584	0.1416	0.0000
OT	0.9715	0.0285	0.0000

Fig. 5. Confusion matrices and the corresponding AUC-ROC plots of the annotation ratio of 0.25.



Ours

Actual \ Predicted	BG	SE	OT
BG	0.9856	0.0079	0.0065
SE	0.3442	0.6282	0.0276
OT	0.4005	0.0670	0.5326

SupSeg

Actual \ Predicted	BG	SE	OT
BG	0.9851	0.0077	0.0071
SE	0.3444	0.6080	0.0475
OT	0.3254	0.1515	0.5231

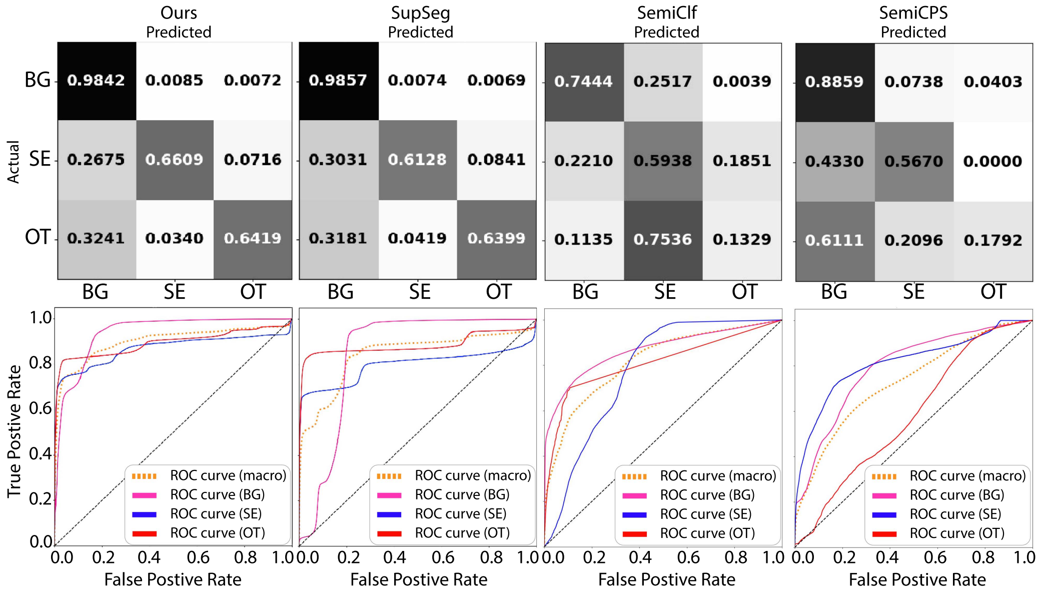
SemiClf

Actual \ Predicted	BG	SE	OT
BG	0.8530	0.1411	0.0059
SE	0.2510	0.6639	0.0851
OT	0.1491	0.8009	0.0501

SemiCPS

Actual \ Predicted	BG	SE	OT
BG	0.9155	0.0602	0.0243
SE	0.6450	0.3295	0.0255
OT	0.5800	0.2085	0.2115

Fig. 6. Confusion matrices and the corresponding AUC-ROC plots of the annotation ratio of 0.30.



Ours

Actual \ Predicted	BG	SE	OT
BG	0.9842	0.0085	0.0072
SE	0.2675	0.6609	0.0716
OT	0.3241	0.0340	0.6419

SupSeg

Actual \ Predicted	BG	SE	OT
BG	0.9857	0.0074	0.0069
SE	0.3031	0.6128	0.0841
OT	0.3181	0.0419	0.6399

SemiClf

Actual \ Predicted	BG	SE	OT
BG	0.7444	0.2517	0.0039
SE	0.2210	0.5938	0.1851
OT	0.1135	0.7536	0.1329

SemiCPS

Actual \ Predicted	BG	SE	OT
BG	0.8859	0.0738	0.0403
SE	0.4330	0.5670	0.0000
OT	0.6111	0.2096	0.1792

Fig. 7. Confusion matrices and the corresponding AUC-ROC plots of the annotation ratio of 0.35.

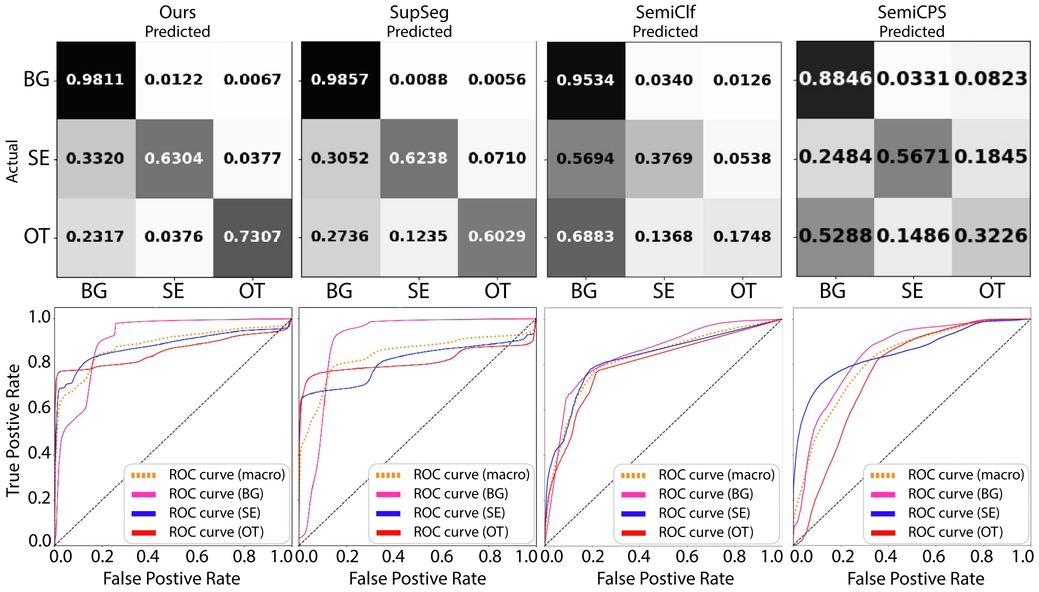


Fig. 8. Confusion matrices and the corresponding AUC-ROC plots of the annotation ratio of 0.40.

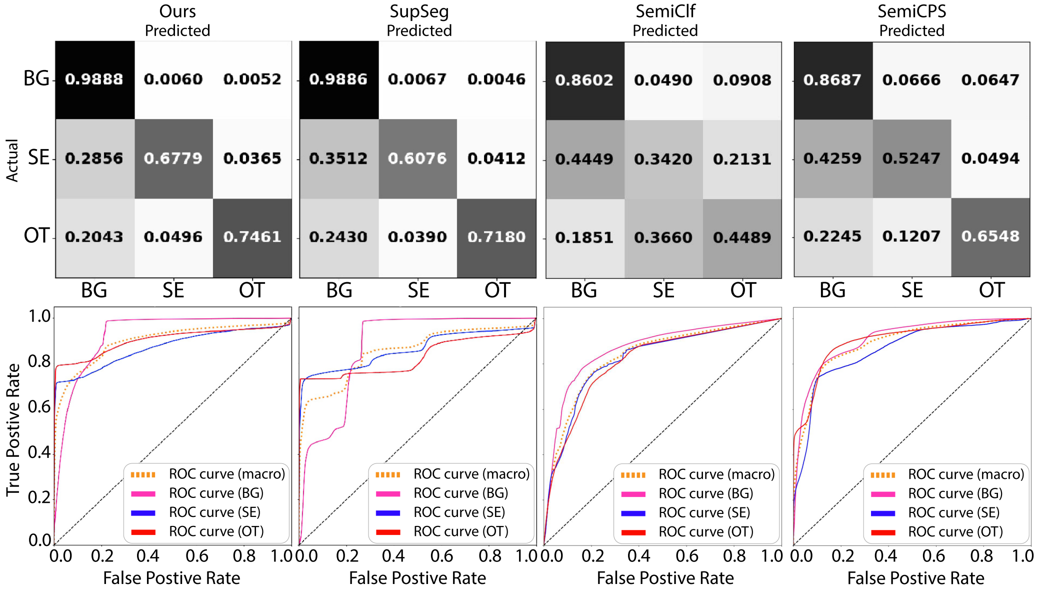


Fig. 9. Confusion matrices and the corresponding AUC-ROC plots of the annotation ratios of 1.00.

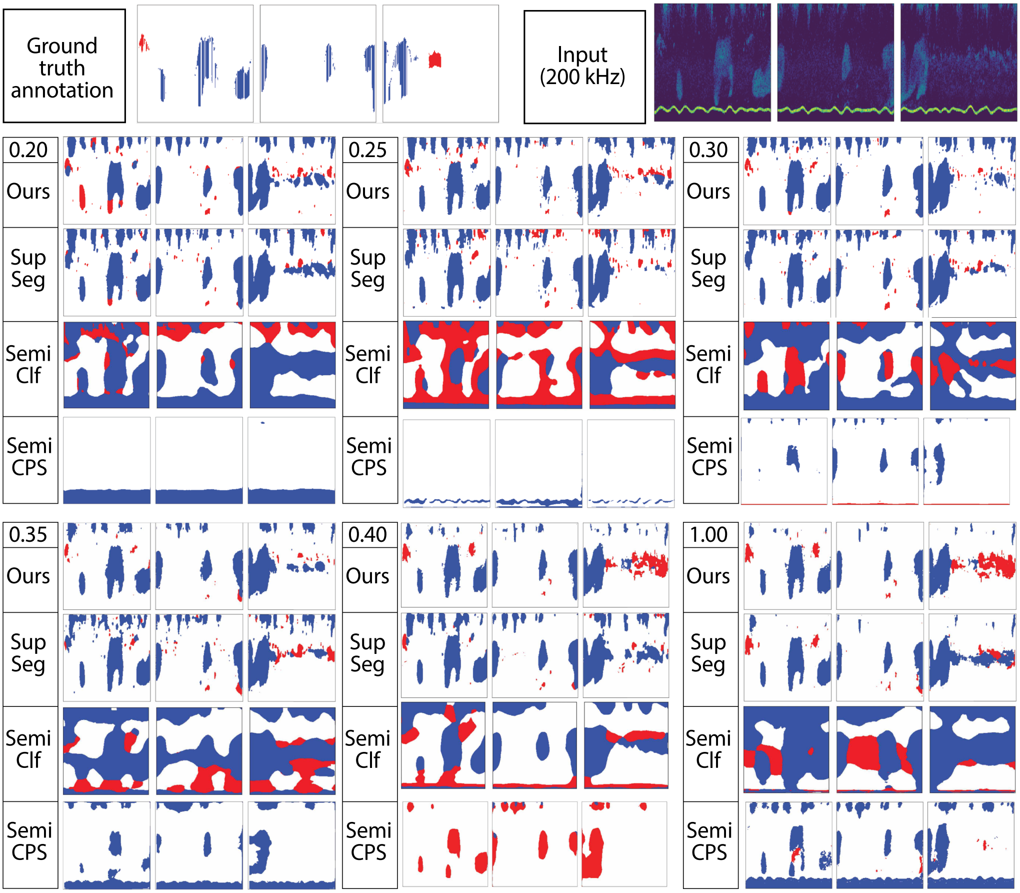


Fig. 10. Prediction maps of the test echosounder data with respect to the annotation ratios. The colors in the annotation map indicate the classes: background (BG) in white, other fish species (OT) in red, and sandeel (SE) in blue.

C. Comparison to Fully Supervised Method (SupSeg)

We compare the result of our method to SupSeg [5], to investigate how the unsupervised clustering objective and the unannotated data improve the predictive performance. Overall, our proposed method outperforms SupSeg through the entire annotation ratios for the entire AUC-ROC values and the SE and OT class accuracies in Table III. The results indicate that the unsupervised clustering objective improves the performance of the segmentation task by effectively exploiting the structured representation from both the unannotated data and the available annotated data.

Note that our proposed method outperforms SupSeg for the annotation ratio of 1.00 (fully supervised case). With this result, we argue that our proposed method is generic and outperforms the conventional fully supervised learning methods, such as SupSeg. Alternating two objective functions are applicable to the fully supervised case, which facilitates the interconnection of the two objectives to make good use of the annotated data based on the clustering structure. By the iteration, the datapoints in each cluster gradually share the dominant class annotation, and eventually have the same class prediction, approximating the decision boundaries that SupSeg achieves to some extent.

In Table IV, we find two inconsistent cases for the annotation ratios of 0.35 and 0.40, where SupSeg achieves greater Kappa and F1 scores. However, we argue that this result does not undermine the robustness of our proposed method. Instead, we believe that SupSeg is biased to make more predictions for the BG class, where the bias is related to a severe class imbalance in the training data, especially in the increased part of the annotated data. The prediction accuracy of the BG class for these annotation ratios validates our reasoning, where SupSeg achieves better accuracy than our proposed method for these annotation ratios (SupSeg 0.9857, ours 0.9842 with the annotation ratio of 0.35; SupSeg 0.9811, ours 0.9857 with the annotation ratio of 0.40).

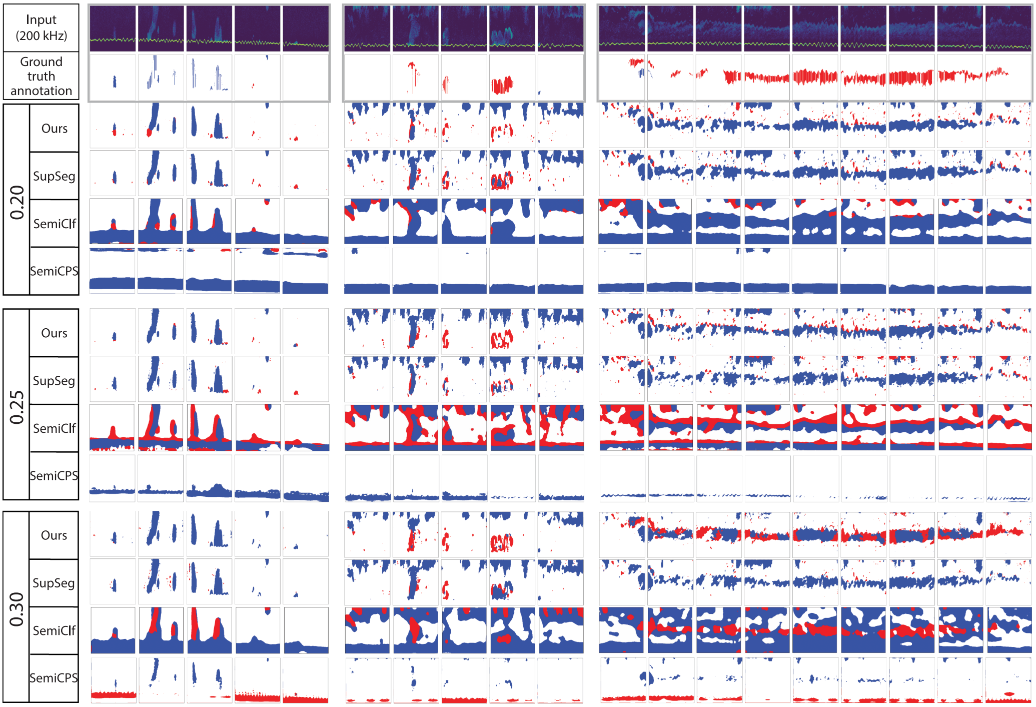


Fig. 11. Prediction maps of the test echosounder data with the annotation ratio of 0.20, 0.25, and 0.30. The colors in the annotation map indicate the classes: background (BG) in white, other fish species (OT) in red, and sandeel (SE) in blue. (a) Case where the SE class is dominant, whereas (b) and (c) show the case where the OT class is dominant.

On the other hand, the prediction accuracies of two fish classes do not seem to increase as much as it increases in our method (ours 0.6609, SupSeg 0.6128 with the annotation ratio of 0.35 and the SE class accuracy; ours 0.6304, SupSeg 0.6238 with the annotation ratio of 0.40 and the SE class accuracy; ours 0.6419, SupSeg 0.6399 with the annotation ratio of 0.35 and the OT class accuracy; ours 0.7307, SupSeg 0.6029 with the annotation ratio of 0.40 and the OT class accuracy). Through visual inspection of the annotated part of the training data, we are able to obtain other grounds for our argument.

When performing the visual inspection of the increased part of the training set between the annotation ratio of 0.30 and 0.35, where ten input-annotation data pairs are increased, we discover that five out of ten data pairs consist of only BG class pixels without any fish class pixel. Analogously, we discover that six out of ten data pairs consist of only BG class pixels without any fish class pixel between the annotation ratio of 0.35 and 0.40. For the entire training data, the case that no fish intensity pixels are obtained in the input takes about 20% of the training data on average. Hence, we argue that the class imbalance found with these annotation ratios is more severe than the other cases and causes the prediction bias towards the BG class for the SupSeg case.

D. Confusion Matrix and ROC Curve

Figs. 4–9 compare our proposed method to other comparison models using confusion matrices and ROC curves. When comparing the diagonal components of the confusion matrices visually, our proposed method shows more distinct diagonal components than the other models. This implies that: 1) our proposed method can be seen to outperform the comparison model in terms of the class accuracy as illustrated in Table III; 2) our proposed method also achieves lower false-positive rates within fish classes compared to other models when having a deeper look at the diagonal components of the SE and OT classes (second and third row and column). For example, comparing the false-positive rate of SE prediction of the OT class ground truth, shown in the second column and the third row of the confusion matrices, ours achieves lower false-positive rates throughout the semisupervised setups. This result is consistent with the false-positive rate of OT prediction of the SE class ground truth, shown in the third column and the second row of the matrices.

The ROC curve shows the tradeoff between true-positive and false-positive rates. The curves indicate that segmentation models with curves closer to the top-left corner perform better,

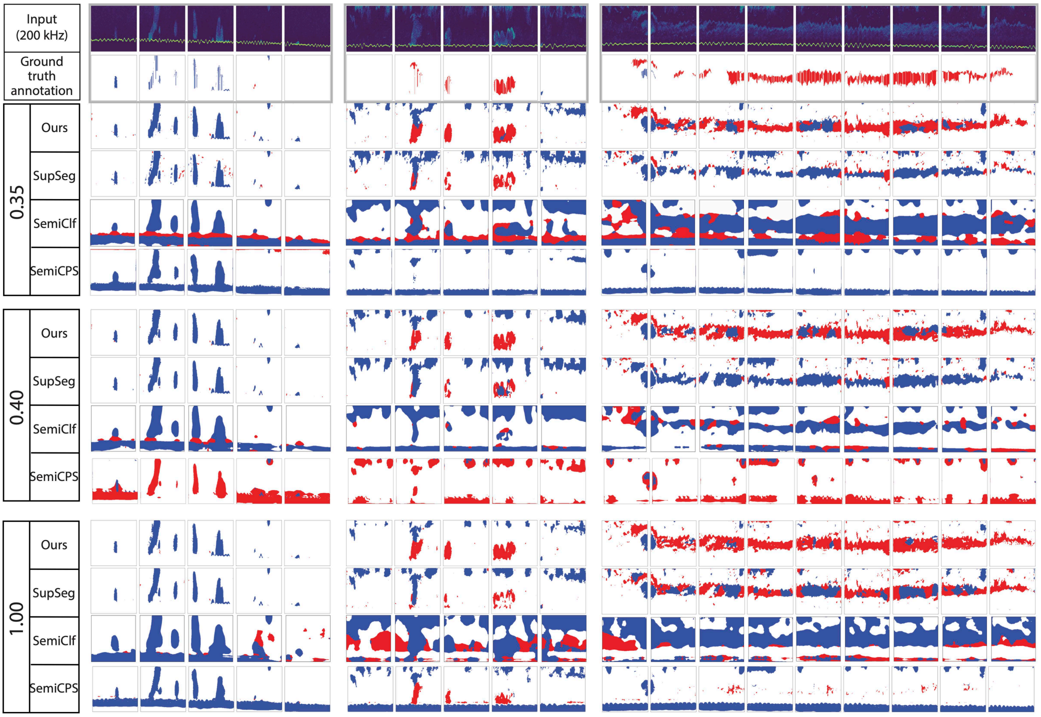


Fig. 12. Prediction maps of the test echosounder data with the annotation ratio of 0.35, 0.40, and 1.00. The colors in the annotation map indicate the classes: background (BG) in white, other fish species (OT) in red, and sandeel (SE) in blue. (a) Case where the SE class is dominant, whereas (b) and (c) show the case where the OT class is dominant.

resulting in greater area under the curve (AUC) as depicted in Table III. The results in the curves and the AUC values validate the outperforming result of our method.

VI. Conclusion

In this article, we propose a novel semisupervised deep learning method for semantic segmentation of echosounder data. Our method considerably reduces the dependence on the annotated data, achieving comparable results with the fully supervised segmentation method [5], by leveraging 40% of the annotated data in addition to unannotated data. Our method also outperforms the other semisupervised methods for echosounder data [22], [71]. Our methodological novelty is to take advantage of deep clustering to exploit the underlying structure of the training data regardless of the annotation in a semisupervised learning scheme. In addition, our method is end-to-end and mini-batch trainable, and regulates the class imbalance based on the model prediction without leveraging the annotated part of data. The rigorous and extensive experiments validate the robustness of the proposed method, where various performance measures are introduced.

Our proposed method is generic and applicable to other fish species with a small amount of annotated echosounder data. To the best of our knowledge, this is the first semisupervised semantic segmentation article for the echosounder data analysis based on deep learning. The promising results imply that our proposed method can reduce the expensive costs required for the annotation. The performance can be improved by utilizing semantic information, e.g., a simple classifier that can exclude the background class pixels when collecting the echosounder data.

In future work, we intend to explore the uncertainty of the segmentation results to improve the interpretability of the model prediction. As a further example of future work, we intend to extend our method to take the uncertainty into account to create more crisp and clear decision boundaries among the clusters when the pseudolabels are created.

References

[1] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[2] T. Zhou, L. Li, X. Li, C.-M. Feng, J. Li, and L. Shao, “Group-wise learning for weakly supervised semantic segmentation,” *IEEE Trans. Image Process.*, vol. 31, pp. 799–811, 2022.

[3] C. Ge, H. Sun, Y.-Z. Song, Z. Ma, and J. Liao, “Exploring local detail perception for scene sketch semantic segmentation,” *IEEE Trans. Image Process.*, vol. 31, pp. 1447–1461, 2022.

[4] W.-J. Lee and T. K. Stanton, “Statistics of broadband echoes: Application to acoustic estimates of numerical density of fish,” *IEEE J. Ocean. Eng.*, vol. 41, no. 3, pp. 709–723, Jul. 2016.

[5] O. Brautaset et al., "Acoustic classification in multifrequency echosounder data using deep convolutional neural networks," *ICES J. Mar. Sci.*, vol. 77, no. 4, pp. 1391–1400, 2020.

[6] P. L. D. Roberts, J. S. Jaffe, and M. M. Trivedi, "Multiview, broadband acoustic classification of marine fish: A machine learning framework and comparative analysis," *IEEE J. Ocean. Eng.*, vol. 36, no. 1, pp. 90–104, Jan. 2011.

[7] R. J. Korneliussen, "Acoustic target classification," *Coop. Res. Rep. - Int. Counc. Explor. Sea*, vol. 344, pp. 1–104, 2018.

[8] J. Simmonds and D. N. MacLennan, *Fisheries Acoustics: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2008.

[9] R. J. Korneliussen, Y. Heggelund, G. J. Macaulay, D. Patel, E. Johnsen, and I. K. Eliassen, "Acoustic identification of marine species using a feature library," *Methods Oceanogr.*, vol. 17, pp. 187–205, 2016.

[10] M. Woillez, P. Ressler, C. Wilson, and J. Horne, "Multifrequency species classification of acoustic-trawl survey data using semi-supervised learning with class discovery," *J. Acoust. Soc. Amer.*, vol. 131, no. 2, pp. 184–190, 2012.

[11] M. Peña, "Robust clustering methodology for multi-frequency acoustic data: A review of standardization, initialization and cluster geometry," *Fisheries Res.*, vol. 200, pp. 49–60, 2018.

[12] R. Proud et al., "Automated classification of schools of the silver cyprinid Rastrineobola argentea in Lake Victoria acoustic survey data using random forests," *ICES J. Mar. Sci.*, vol. 77, no. 4, pp. 1379–1390, 2020.

[13] S. M. Gugele, M. Widmer, J. Baer, J. T. DeWeber, H. Balk, and A. Brinker, "Differentiation of two swim bladdered fish species using next generation wideband hydroacoustics," *Sci. Rep.*, vol. 11, no. 1, pp. 1–10, 2021.

[14] D. Demer et al., "Evaluation of a wideband echosounder for fisheries and marine ecosystem science," *Coop. Res. Rep. - Int. Counc. Explor. Sea*, vol. 336, pp. 1–69, 2017.

[15] P. Baldi, *Deep Learning in Science*. Cambridge, U.K.: Cambridge Univ. Press, 2021.

[16] A. Ordoñez, I. Utseth, O. Brautaset, R. Korneliussen, and N. O. Handegard, "Evaluation of echosounder data preparation strategies for modern machine learning models," *Fisheries Res.*, vol. 254, 2022, Art. no. 106411.

[17] X. Luo, X. Qin, Z. Wu, F. Yang, M. Wang, and J. Shang, "Sediment classification of small-size seabed acoustic images using convolutional neural networks," *IEEE Access*, vol. 7, pp. 98331–98339, 2019.

[18] T. P. Marques et al., "Instance segmentation-based identification of pelagic species in acoustic backscatter data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4378–4387.

[19] S. C. Lowe et al., "Echofilter: A deep learning segmentation model improves the automation, standardization, and timeliness for post-processing echosounder data in tidal energy streams," *Front. Mar. Sci.*, vol. 9, no. 867857, pp. 1–21, 2022.

[20] O. Chapelle et al., "Semi-supervised learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 542–542, 2006.

[21] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11557–11568.

[22] C. Choi et al., "Semi-supervised target classification in multi-frequency echosounder data," *ICES J. Mar. Sci.*, vol. 78, no. 7, pp. 2615–2627, 2021.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2015, pp. 234–241.

[24] T. Glasmachers, "Limits of end-to-end learning," in *Proc. Asian Conf. Mach. Learn.*, 2017, pp. 17–32.

[25] Y. Li et al., "Learning dynamic routing for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8553–8562.

[26] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, 2020.

[27] Q. Liu, M. Kampffmeyer, R. Jenssen, and A.-B. Salberg, "Dense dilated convolutions' merging network for land cover classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6309–6320, Sep. 2020.

[28] A. Vali, S. Comai, and M. Matteucci, "Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review," *Remote Sens.*, vol. 12, no. 15, 2020, Art. no. 2495.

[29] L. T. Luppino et al., "Deep image translation with an affinity-based change prior for unsupervised multimodal change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4700422.

[30] X. Yu, J. Fan, J. Chen, P. Zhang, Y. Zhou, and L. Han, "NestNet: A multiscale convolutional neural network for remote sensing image change detection," *Int. J. Remote Sens.*, vol. 42, no. 13, pp. 4898–4921, 2021.

[31] S. Hansen et al., "Unsupervised supervoxel-based lung tumor segmentation across patient scans in hybrid PET/MRI," *Expert Syst. Appl.*, vol. 167, 2021, Art. no. 114244.

[32] M. A. Naser and M. J. Deen, "Brain tumor segmentation and grading of lower-grade glioma using deep learning in MRI images," *Comput. Biol. Med.*, vol. 121, 2020, Art. no. 103758.

[33] D. Feng et al., "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[34] K. Wickstrøm, M. Kampffmeyer, and R. Jenssen, "Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps," *Med. Image Anal.*, vol. 60, 2020, Art. no. 101619.

[35] D. Jha et al., "Real-time polyp detection, localization and segmentation in colonoscopy using deep learning," *IEEE Access*, vol. 9, pp. 40 496–40 510, 2021.

[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[37] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

[38] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *Int. J. Multimedia Inf. Retrieval*, vol. 7, no. 2, pp. 87–93, 2018.

[39] N. Daan, P. Bromley, J. Hislop, and N. Nielsen, "Ecology of north sea fish," *Netherlands J. Sea Res.*, vol. 26, no. 2–4, pp. 343–386, 1990.

[40] R. W. Furness, "Management implications of interactions between fisheries and sandeel-dependent seabirds and seals in the north sea," *ICES J. Mar. Sci.*, vol. 59, no. 2, pp. 261–269, 2002.

[41] E. Johnsen, G. Rieucau, E. Ona, and G. Skaret, "Collective structures anchor massive schools of lesser sandeel to the seabed, increasing vulnerability to fishery," *Mar. Ecol.: Prog. Ser.*, vol. 573, pp. 229–236, 2017.

[42] D. N. MacLennan, P. G. Fernandes, and J. Dalen, "A consistent approach to definitions and symbols in fisheries acoustics," *ICES J. Mar. Sci.*, vol. 59, no. 2, pp. 365–369, 2002.

[43] R. Kloser, T. Ryan, P. Sakov, A. Williams, and J. Koslow, "Species identification in deep water using multiple acoustic frequencies," *Can. J. Fish. Aquatic Sci.*, vol. 59, no. 6, pp. 1065–1077, 2002.

[44] R. J. Korneliussen and E. Ona, "Synthetic echograms generated from the relative frequency response," *ICES J. Mar. Sci.*, vol. 60, no. 3, pp. 636–640, 2003.

[45] D. G. Reid, "Report on echo trace classification," *Coop. Res. Rep. - Int. Counc. Explor. Sea*, vol. 238, pp. 1–107, 2000.

[46] N. O. Handegard and D. Tjøstheim, "The sampling volume of trawl and acoustics: Estimating availability probabilities from observations of tracked individual fish," *Can. J. Fisheries Aquatic Sci.*, vol. 66, no. 3, pp. 425–437, 2009.

[47] T. K. Stanton et al., "On acoustic estimates of zooplankton biomass," *ICES J. Mar. Sci.*, vol. 51, no. 4, pp. 505–512, 1994.

[48] D. A. Demer and S. G. Conti, "Reconciling theoretical versus empirical target strengths of krill: Effects of phase variability on the distorted-wave born approximation," *ICES J. Mar. Sci.*, vol. 60, no. 2, pp. 429–434, 2003.

[49] E. Johnsen, R. Pedersen, and E. Ona, "Size-dependent frequency response of sandeel schools," *ICES J. Mar. Sci.*, vol. 66, no. 6, pp. 1100–1105, 2009.

[50] M. Barange, "Acoustic identification, classification and structure of biological patchiness on the edge of the Agulhas bank and its relation to frontal features," South *Afr. J. Mar. Sci.*, vol. 14, no. 1, pp. 333–347, 1994.

[51] J. Coetzee, "Use of a shoal analysis and patch estimation system (SHAPES) to characterise sardine schools," *Aquat. Living Resour.*, vol. 13, no. 1, pp. 1–10, 2000.

[52] S. Aronica et al., "Identifying small pelagic mediterranean fish schools from acoustic and environmental data using optimized artificial neural networks," *Ecol. Inform.*, vol. 50, pp. 149–161, 2019.

[53] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[54] M. Jabi, M. Pedersoli, A. Mitiche, and I. B. Ayed, "Deep clustering: On the link between discriminative models and k-means," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1887–1896, Jun. 2021.

[55] D.-H. Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," *Proc. Int. Conf. Mach. Learn.*, vol. 3, no. 2, 2013, pp. 896–902.

[56] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[57] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[58] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, 1987.

[59] T. Lin, L. Kong, S. Stich, and M. Jaggi, “Extrapolation for large-batch training in deep learning,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6094–6104.

[60] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[61] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, “Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1–9.

[62] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[63] M. Hyun, J. Jeong, and N. Kwak, “Class-imbalanced semi-supervised learning,” in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–16.

[64] L.-Z. Guo et al., “Learning from imbalanced and incomplete supervision with its application to ride-sharing liability judgment,” in *Proc. 27th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 487–495.

[65] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang, “CReST: A class-rebalancing self-training framework for imbalanced semi-supervised learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10857–10866.

[66] M. L. McHugh, “Interrater reliability: The kappa statistic,” *Biochem. Med.*, vol. 22, no. 3, pp. 276–282, 2012.

[67] A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[68] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[69] L. Prechelt, “Early stopping-but when?,” in *Neural Networks: Tricks of the Trade*. New York, NY, USA: Springer, 1998, pp. 55–69.

[70] A. Paszke et al., “Pytorch: An imperative style, high-performance deep learning library,” in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[71] X. Chen, Y. Yuan, G. Zeng, and J. Wang, “Semi-supervised semantic segmentation with cross pseudo supervision,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2613–2622.

Changkyu Choi received the bachelor’s degree in electronic and electrical engineering from Sungkyunkwan University, Suwon, South Korea, in 2011, the first master’s degree in user experience (UX) design from the Graduate School of Information, Yonsei University, Seoul, in 2016, the second master’s degree in machine learning in 2019 from the Department of Physics and Technology, UiT The Arctic University, Tromsø, Norway, where he is currently working toward the Ph.D. degree in physics.

He is currently an Assistant Professor with UiT The Arctic University. From 2011 to 2013, he was with Samsung Electronics, Seoul, South Korea, where he primarily worked on researching and developing mobile communication devices. In 2022, he worked as a Researcher with the Norwegian Computing Center, Oslo, Norway.

His research interests include computer vision and deep learning, and in the application of deep learning to marine image analysis.

Mr. Choi is a Design Chair of the annual Northern Lights Deep Learning (NLDL) Conference.

Michael Kampffmeyer (Member, IEEE) received the Ph.D. degree in physics from UiT The Arctic University, Tromsø, Norway, in 2018.

He is an Associate Professor and the Head of the Machine Learning Group with UiT The Arctic University. He is also a Senior Research Scientist II with the Norwegian Computing Center, Oslo, Norway. He has had long-term research stays with the Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA, and the Berlin Center for Machine Learning, Technical University of Berlin, Berlin, Germany. His research interests include medical image analysis, explainable AI, and learning from limited labels (e.g., clustering, few/zero-shot learning, domain adaptation, and self-supervised learning).

Dr. Kampffmeyer is a General Chair of the annual Northern Lights Deep Learning (NLDL) Conference. For more details visit https://sites.google.com/view/michaelkampffmeyer/

Nils Olav Handegard (Member, IEEE) received the Dr. Sci. (Ph.D.) degree in applied mathematics from the University of Bergen, Bergen, Norway, in 2004.

He is currently a Principal Research Scientist with the Institute of Marine Research, Bergen, Norway. He has been part of the Science Leadership group in the International Council for Exploration of the Sea (ICES), and has contributed to a range of working groups, including leading the Fisheries Acoustics Science and Technology group (WGFAST) and the steering group overseeing ICES coordinated scientific surveys. He has had longer term Visiting Scholar stays with the University of Washington, Seattle, WA, USA, and with Princeton University, Princeton, NJ, USA, working on acoustic sensors to observe fish behavior. His research interests include applications of new methodology and data processing methods within marine ecology and fisheries oceanography, including machine learning and deep learning algorithms.

Dr. Handegard is leading the CRIMAC Center, a Norwegian Centre for research-innovation funded by the research council of Norway.

Arnt-Børre Salberg (Member, IEEE) received the diploma degree in applied physics and the Dr.Sci. degree in physics from the University of Tromsø, Tromsø, Norway, in 1998 and 2003, respectively.

He is currently a Senior Research Scientist in earth observation with Norwegian Computing Center, Oslo, Norway. From February 2003 to December 2005, he had a Postdoctoral and research position with the Institute of Marine Research, Tromsø. From December 2005 to October 2008, he was the Head of Research and Development with Dolphiscan AS, Moelv, Norway. Since October 2008, he has been with Norwegian Computing Center, Oslo, Norway. From August 2001 to June 2002, he was a Visiting Researcher with the U.S. Army Research Laboratory, Adelphi, MD, USA. His research interests include earth observation, computer vision, machine learning, and statistics.

Robert Jenssen (Senior Member, IEEE) received the Dr. Sci. (Ph.D.) in physics from UiT The Arctic University of Norway, Tromsø, Norway, in 2005.

He is the Director of Visual Intelligence, a Norwegian Centre for research-based innovation funded by the Research Council of Norway and consortium partners. Visual Intelligence solves research challenges in deep learning to advance image analysis. He is a Professor and Founder of the Machine Learning Group, UiT The Arctic University of Norway. He is in addition a part time Professor with the Pioneer AI Centre, University of Copenhagen, Copenhagen, Denmark, and an Adjunct Professor with the Norwegian Computing Center, Oslo, Norway. He has had long-term research stays with the University of Florida, Gainesville, FL, USA, Technical University of Denmark, Kongens Lyngby, Denmark, and Technical University of Berlin, Berlin, Germany. His research interests include e-science, data management, data processing, neural networks, graph and kernel-based learning, and in health and industrial applications of machine learning.

Dr. Jenssen has been on the IEEE MLSP TC and on the Governing Board of IAPR. He is an Editor for the journal *Pattern Recognition* and a Member of the ELLIS Cph unit. He is the General Chair of the annual Northern Lights Deep Learning (NLDL) Conference.

11 | Paper III

Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Pattern Recognition 00 (2023) 1–21

Pattern Recognition

Deep Deterministic Information-Bottleneck Explainability on Marine Image Data

Changkyu Choi[a,b], Shujian Yu[a,b,e], Michael Kampffmeyer[a,b,d], Arnt-Børre Salberg[a,d], Nils Olav Handegard[a,c], Robert Jenssen[a,b,d]

[a] *SFI Visual Intelligence, PO Box 6050 Langnes, N-9037 Tromsø, Norway*
[b] *UiT The Arctic University of Norway, PO Box 6050 Langnes, N-9037 Tromsø, Norway*
[c] *Norwegian Institute of Marine Research, PO Box 1870 Nordnes, NO-5817 Bergen, Norway*
[d] *Norwegian Computing Center, PO Box 114 Blindern, NO-0314 Oslo, Norway*
[e] *Vrije Universiteit Amsterdam, de Boelelaan 1081a, 1081HV Amsterdam, Netherlands*

Abstract

We propose DIB-X, a generic self-explainable deep learning approach, tested in marine environment monitoring applications. Our method generates explanations through optimization, adhering to the principles of minimal, sufficient, and interactive explanations. The minimality and sufficiency principles are rooted within the information bottleneck framework. Distinctly, DIB-X directly quantifies the minimality principle using the recently proposed matrix-based Rényi's α-order entropy functional, circumventing the need for variational approximation and distributional assumption. The interactivity principle is realized by incorporating existing domain knowledge as prior explanations, fostering explanations that align with established domain understanding. Empirical results on two marine environment monitoring datasets with different modalities reveal that our approach primarily provides improved explainability, with the added advantage of enhanced classification performance. The source code for DIB-X is publicly accessible at *github.com/SFI-Visual-Intelligence/DIB-X*.

Keywords: Explainable deep learning, self-explainability, information-bottleneck, matrix-based Rényi's α-order entropy functional, multi-frequency echosounder data, seal pup images on sea ice

1. Introduction

The significance of monitoring the marine environment is paramount, as the ocean plays a vital role in supporting life on Earth [1]. Deep learning has transformed the field of marine environmental monitoring by enabling automated visual monitoring that was previously unfeasible due to the extensive resources, time, and expertise needed [2, 3].

Despite the advancements in automated visual monitoring, there are some challenges in applying deep learning-based systems to real-world situations. First, deep learning systems exhibit a level of complexity that makes them akin to black boxes, which complicates the understanding of the decision-making process behind their outcomes [4]. Furthermore, the promising results demonstrated so far have primarily been achieved under 'sandbox conditions', a controlled test environment within computer systems where new techniques can be safely executed [5].

Given the potential negative consequences of inaccurate marine environment monitoring, it is prudent to ensure that deep learning systems provide human-understandable explanations for their decisions, thereby fostering trust in their outcomes [6]. This approach enables the utilization of deep learning-based monitoring systems as a supplement to human decision-making rather than a replacement, ultimately enhancing the overall monitoring process [7].

In this context, it is evident that the explainability of deep learning has emerged as one of the critical topics in the field of computational intelligence [5, 7]. Its primary objective is to develop new decision-making systems that can additionally provide human-understandable explanations, making their decisions more trustworthy [6].

The growing interest in explainability methods within the field of marine environmental monitoring is observed in [8]. However, further research and investigation are necessary to fully realize their potential benefits. As explainability methods continue to advance and gain credibility, it is anticipated that deep learning-based systems will contribute more significantly to marine environmental monitoring in a variety of scenarios [9]. These include the analysis of images collected by unmanned aerial vehicles [10] and underwater images from multi-frequency echosounders [11, 12].

In this paper, we propose DIB-X, a novel explainable deep learning method, which stands for deep deterministic information bottleneck explainability. Our proposed method is evaluated using two marine environment monitoring image datasets with different modalities, including multi-frequency echosounder data [11, 12] and the images of seal pups on sea ice [10].

Notably, methodological genericity is a key aspect of DIB-X, enabling applicability across various domains beyond marine environmental monitoring. Leveraging the information bottleneck (IB) framework [13], DIB-X addresses two contrasting principles for the explanation, including *sufficiency* with respect to the output and *minimality* concerning the input. DIB-X seeks to learn the optimal latent representation as an explanation by balancing the trade-off between these two principles.

Distinctly, as a crucial feature of DIB-X, we incorporate the matrix-based Rényi's α-order entropy functional [14] to the IB framework, resulting in a more robust explainability method compared to existing IB-based approaches [15, 16]. This advanced entropy measure [14] circumvents the need for estimating the probability density of variables, streamlining the neural network's decision-making explanation process.

Additionally, we present an extension of the proposed DIB-X that integrates available domain knowledge during the learning of explanations, making the resulting explanations more *interactive* within the specific domain. Principles such as *sufficiency*, *minimality*, and *interactivity* are incorporated into the objective function through mathematical formulation. This enables a self-explainable learning scheme that provides explanations alongside predictions during optimization [6, 17].

Empirically, DIB-X demonstrates trustworthiness by prioritizing the clarity of its explanations while maintaining enhanced classification performance. Our main contributions can be summarized as follows:

- To propose a novel explainable deep learning method that addresses *sufficient* but *minimal* explanation based on the information-bottleneck framework [13].

- To introduce, for the first time, the matrix-based R'enyi's α-order entropy functional [14] to explainability methods.

- To formulate the proposed method so that it performs within an end-to-end and self-explainable scheme.

- To extend the proposed method, enabling it to integrate domain knowledge during the learning of explanations, making the explanations more *interactive*.

- To demonstrate the applicability of the proposed method to real-world marine environment monitoring data with different modalities.

2. Related work

The field of model explainability methods has seen rapid growth in recent years. The aim is to provide insight into how complex models, such as neural networks, make predictions [18]. Among the explainability methods, attribution methods [19, 18] have become a common choice when dealing with image data. The attributions, denoted as M in this paper, explain the model behaviour in the input pixel space by assigning relevance scores, which highlight salient areas relevant to network decisions [16].

Earlier explainability methods provide *a-posteriori* explanations leveraging pretrained models considering the models as a black-box [20, 21, 19, 22]. One line of research is represented by the gradient-based method, where the attribution is calculated by the gradients or reverse propagation to the input space. This includes Grad-CAM [19], LRP [20], and DeepLIFT [21], to name a few. Grad-CAM [19] uses the gradients of logits, flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image. Although the gradient-based methods are relatively straightforward to implement in the given network architecture, the explanation quality is to some extent limited due to gradient shattering property connected to gradient approaches, and the visualizations based on gradient-based methods often contain falsely perceptual regions in addition to a coarse representation.

Though gradient-based methods are relatively simple to implement within a given network architecture, their explanation quality is somewhat limited due to the gradient shattering property associated with these approaches. Consequently, the visualizations based on gradient-based methods frequently exhibit falsely perceptual regions and a coarse representation.

Another line of *a-posteriori* explanation research is the perturbation-based method. This approach aims to observe the output changes by processing a set of perturbed images, where each has an occlusion of a different region in the image. This includes LIME [22], Occlusion [23], and RISE [24]. A well-known method LIME [22] first employs occlusions of superpixels from the original image to synthesize a number of neighboring image instances. The synthesized instances and the outcomes are used to fit a linear model where the coefficients of the linear model explain the contributions of occluded features. Perturbation-based methods are known for providing robust and reliable explanations in the input space. Nevertheless, the number or resolution of input perturbations that can be sampled is limited by practical constraints due to the rapid increase in combinatorial complexity.

Beyond the *a-posteriori* explanation methods, recent works have proposed self-explaining methods [5, 17], which aim to integrate explainability factors, such as intelligibility [5], coherence [17], and minimality [15], with the learning process to account for what constitutes a good explanation. While the exact formulation may vary depending on the method and the application, a common approach is to mathematically formulate such factors and add them to the objective function in addition to a term seeking for input-output relevance, such as a cross-entropy. This differs from *a-posteriori* methods in that their pretrained network often has learned with a focus only on input-output relevance.

To facilitate the integration of the explainability factors, the self-explaining methods tend to separate the network architecture to learn the individual representation for the explanation and prediction. For example, a general self-explaining model SENN [17] consists of three separate network modules. A concept encoder transforms the input into explainable basis representations, an input-dependent parametrizer generates relevance scores regarding the basis representations, and an aggregator combines both the representations and the scores to produce a prediction. Due to the higher modelling capacity of the neural networks, the self-explaining methods can provide more informative, understandable, and transparent explanations [17, 6]. This is also an effort to change the black-box neural networks to a so-called glass-box [25].

Building on numerous measures of information quantities and learning principles based on information theory [26, 27, 13, 28], recent explainable deep learning methods [15, 16, 18, 29] have leveraged the information-bottleneck (IB) framework [13]. The IB framework [13], which aims to find the optimal trade-off between *minimality* and *sufficiency* of information at the latent representation, has gained particular attention in recent years due to its solid theoretical foundation and potential for formulating self-explainable deep learning methods [15]. The DIB-X method proposed in this paper takes further novel steps forward towards IB-based self-explainable deep learning, as elaborated in the next section.

3. The proposed deep deterministic information bottleneck explainability approach

The aim of the IB framework is to learn a latent representation T between an input X and a target Y. The idea is that T should capture a *minimal* amount of information about X while at the same time retaining *sufficient* information about the target Y [30, 15]. This creates a so-called information bottleneck, producing a T which represents the best trade-off between *minimality* and *sufficiency* of information with respect to X and Y, respectively [14].

The deep IB framework implements the IB framework in the context of deep learning, and allows it to take advantage of the powerful feature learning capabilities provided by deep neural networks. This is particularly useful when dealing with high-dimensional and complex data, such as images [28, 14, 15]. This is achieved by maximizing the objective function defined by mutual information, which encourages the network to learn the bottleneck representation T, a compressed representation of the input data X that preserves the relevant information of the target Y.

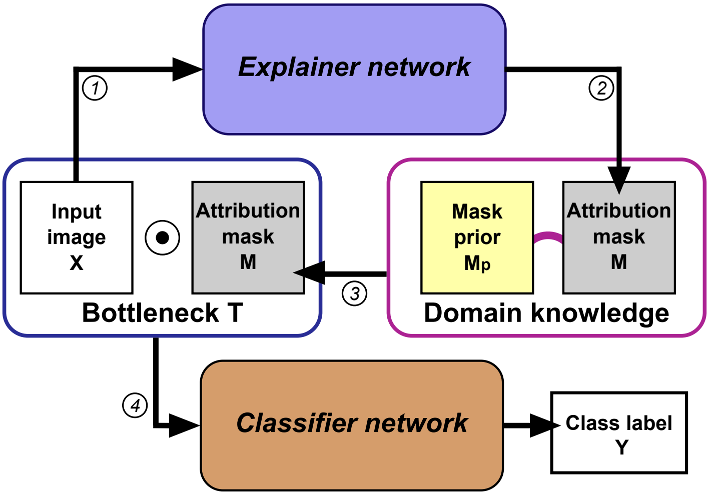


Figure 1. Overview of the proposed DIB-X. The method consists of four steps. ① The input image X is used to create the attribution mask M by being processed by the *explainer* network module. ② If domain knowledge in the form of the mask prior M_p is available, it can be integrated into the attribution mask. ③ The attribution mask is employed to generate the bottleneck representation T, which is obtained by taking the Hadamard product of the mask with the input image X. ④ The *classifier* network module processes the bottleneck representation T to perform the classification task.

3.1. Information-bottleneck and attribution-based explanation

Mutual information is a measure of shared information between two random variables. It plays a crucial role in the IB framework [13] as it defines the learning criteria. The mutual information $I(X;Y)$ between two random variables X and Y is defined as

$$I(X;Y) = \mathbb{E}_{X,Y}[\log \frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}] = H(X) + H(Y) - H(X,Y) = H(Y) - H(Y \mid X), \tag{1}$$

where $p_X(x)$, $p_Y(y)$, and $p_{XY}(x,y)$ are the marginal and joint probability distributions of the variables. The quantities $H(X)$, $H(Y)$, $H(X,Y)$ and $H(Y \mid X)$ are the entropy of X and Y, their joint entropy, and the conditional entropy, respectively. Note that the mutual information $I(X;Y)$ is non-negative and symmetric in the two random variables.

With the mutual information of two variables $I(\cdot\,;\,\cdot)$ as the learning criteria, the general objective function of the IB framework $\mathcal{L}$ is defined as

$$\mathcal{L} = I(T;Y) - \beta I(X;T), \tag{2}$$

where β is a Lagrange multiplier that controls the trade-off between the *sufficiency* and the *minimality* and the aim is to maximize $\mathcal{L}$.

In our proposed work, the latent representation T is learned from image data by optimizing two deep neural network modules connected in series (Figure 1), where the role of the first network is to learn a representation for T. Working with image data, and in line with other relevant literature [18, 29, 16, 15], we define T as the Hadamard product (element-wise multiplication over pixels) between a learnable mask M and the image input X, i.e.,

$$T = M \odot X. \tag{3}$$

The mask M is referred to as an attribution mask and simulates spatial feature removal over the the input image by masking elements of X partially or completely out since each element in M is restricted to have a value between zero and one. Since M acts as an attribution mask to create a bottleneck T, the first network module in essence *explains* the target classification and for this reason we refer to this network as the *explainer*.

The *classifier* serves as the second network module in this setup. The input to the *classifier* is represented by T, and the objective is to maximize the mutual information between T and the target Y, i.e., $I(T;Y)$. It is worth noting that the maximization of the mutual information $I(T;Y)$ can be approximated by minimizing the cross-entropy loss, which will be elaborated upon in the subsequent section. Utilizing this approximation, the *classifier* module prioritizes input-output relevance, while the attribution mask M generated by the *explainer* module controls the amount of information passed to the input of the *classifier* module.

We aim to simultaneously optimize both network modules in an end-to-end fashion with gradient descent. However, this necessitates ways to quantify mutual information in Equation 2. This is the topic for the next section. Since the explanations represented by M will be learned simultaneously as the classification, such a procedure is inherently *self-explainable*.

3.2. Self-explaining IB by deterministic matrix-based Rényi's entropy

Our proposed method takes inspiration from Zhmoginov's work [29] and related IB approaches [15]. [29] presents an IB approach to generate attribution masks M for image classification models, where the idea is to direct model attention away from distracting features and towards features that define the image label. Their model, however, is only evaluated for benchmark image data sets such as MNIST and CIFAR-10, and is based on the variational IB proposed in [28], which is also the case for [15].

The variational IB allows for the approximation of $I(X;T)$ through the variational lower bound principle, which relies on the selection of an appropriate prior distribution [28]. One way to achieve this is to use the prior distribution for the attribution mask M [15]. The Kullback-Leibler (KL) divergence $D_{KL}(p(m|x)\|r(m))$ is used to approximate the lower bound, where $p(m|x)$ is the learned distribution of the attribution mask M and where $r(m)$ is a prior distribution of M. To achieve this, [29] introduces an additional variational autoencoder (VAE) network to formulate the variational IB [28], which reconstructs the masked input T with the selected prior distribution in the latent space of the network.

A critical aspect of this approach is the selection of an appropriate prior distribution, which can significantly impact the quality of the estimation [31]. An alternative approach to avoid the requirement for a prior distribution is the mutual information neural estimator (MINE) [32], which however utilizes an auxiliary network to estimate a lower bound on $I(X;T)$. Indeed, these methods for approximating $I(X;T)$ are valuable tools in IB optimization [33]. However, they still rely on additional mechanisms such as the manual selection of the prior distribution or the inclusion of auxiliary networks, which may impact the quality of the lower bound estimation and pose challenges for practical applications. In order to apply the IB framework to larger networks and real-world data, such as marine image data, a fundamentally novel method is needed to provide a more robust manner for computing mutual information $I(X;T)$.

We, on the other hand and as a novel step, couple our fully self-explainable deep learning-based information bottleneck concept to a recent line of research where direct optimization of the IB objective without any variational approximation has been shown to be successful in the sense of obtaining more robust results compared to the variational approach [30, 34, 14]. This approach to quantify mutual information is deterministic, avoiding variational inference and distributional assumption, and lends itself nicely to learning by gradient descent over mini-batches. We name this new approach deep deterministic information bottleneck explainability (DIB-X). The approach is described below.

In DIB-X, we exploit the relation between mutual information and the entropy, e.g., $I(X;T) = H(X) + H(T) - H(X,T)$. Towards this end, DIB-X quantifies the mutual information using the recently proposed marginal and joint matrix-based Rényi's α-order entropy functional [14]. In this measure, the entropy is quantified directly from the data samples in a mini-batch level via a normalized Gram matrix A^X. The Gram matrix encodes the pairwise relationships among the data samples [14]: Given a set of vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, the normalized Gram matrix $A^X = K^X/tr\left(K^X\right)$ is defined via $K^X \in \mathbb{R}^{n\times n}$ using a real-valued positive definite kernel κ, such that $K^X_{ij} = \kappa\left(\mathbf{x}_i, \mathbf{x}_j\right)$. Here, a common choice for the kernel κ is the radial basis function, e.g., $\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$, where the kernel width σ is a tunable hyperparameter.

With the normalized Gram matrix A^X, the marginal entropy $H_\alpha(A^X)$ is defined as

$$H_\alpha(A^X) = \frac{1}{1-\alpha}\log_2\left(tr\left((A^X)^\alpha\right)\right) = \frac{1}{1-\alpha}\log_2\left(\sum_{i=1}^{n}\lambda_i(A^X)^\alpha\right), \tag{4}$$

where $\alpha \in (0, 1) \cup (1, \infty)$, and $\lambda_i(A^X)$ denotes the i-th eigenvalue of A^X.

The joint entropy $H_\alpha(A^X, A^T)$ is defined with two sets of vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_n\}$ and $T = \{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \cdots, \mathbf{t}_n\}$, where A^T indicates the normalized Gram matrix for T and is defined as

$$H_\alpha(A^X, A^T) = H_\alpha\Big(\frac{A^X \odot A^T}{tr(A^X \odot A^T)}\Big). \tag{5}$$

With this framework, therefore, the mutual information $I(X; T)$ in our proposed DIB-X is

$$I(X; T) = H_\alpha(A^X) + H_\alpha(A^T) - H_\alpha(A^X, A^T). \tag{6}$$

Finally, this enables us to define the a new objective function $\mathcal{L}_{DIB\text{-}X}$ to maximize

$$\mathcal{L}_{DIB\text{-}X} = I(T; Y) - \beta I(X; T) = I(T; Y) - \beta(H_\alpha(A^X) + H_\alpha(A^T) - H_\alpha(A^X, A^T)), \tag{7}$$

where $H_\alpha(A^X)$, $H_\alpha(A^T)$, and $H_\alpha(A^X, A^T)$ are the marginal and joint entropy of the input image X and the masked image T.

Note that maximization of $I(T; Y)$ is approximated by minimizing the conditional entropy $H(Y \mid T)$ in that $I(T; Y)$ is equal to $H(Y) - H(Y \mid T)$, i.e.,

$$\max I(T; Y) = \max \left(H(Y) - H(Y \mid T)\right) = H(Y) + \max\left(- H(Y \mid T)\right) \iff \min H(Y \mid T). \tag{8}$$

Furthermore,

$$H(Y \mid T) \simeq \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p(\mathbf{x},\mathbf{y})}\left[\mathbb{E}_{\mathbf{t}\sim p_\phi(\mathbf{t}|\mathbf{x}_i)}\left[-\log p_\theta(\mathbf{y} \mid \mathbf{t})\right]\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathbf{t}\sim p(\mathbf{t}|\mathbf{x}_i)}\left[-\log p\left(\mathbf{y}_i \mid \mathbf{t}\right)\right] = CE(\hat{Y}, Y), \tag{9}$$

where $CE(\hat{Y}, Y)$ is the cross-entropy loss between the class label Y and prediction $\hat{Y}$ of the images. Hence, the resulting optimization problem may be expressed as

$$\max \mathcal{L}_{DIB\text{-}X} = \max\left[-CE(\hat{Y}, Y) - \beta(H_\alpha(A^X) + H_\alpha(A^T) - H_\alpha(A^X, A^T))\right]. \tag{10}$$

3.3. Integrating domain knowledge through prior M_p

Our proposed DIB-X method effectively integrates domain knowledge into the learning process. For example, we use class labels Y as a source of domain knowledge and exploit them in the classifier module to maximize the mutual information $I(T; Y)$. The explainer module also learns the class labels by backpropagating the gradient of $I(T; Y)$ through the bottleneck representation T.

The attribution mask M, integrated into the explainer module, identifies the relevant region in the input X that correspond to the class labels Y. If local information that can provide a prior belief in estimating the attribution M is readily available as an additional source of domain knowledge, it should also be integrated into the learning process. Hence, an extension of our proposed method with a given attribution mask prior M_p can be expressed as

$$\max \mathcal{L}_{DIB\text{-}X} = \max\left[-CE(\hat{Y}, Y) - \beta(H_\alpha(A^X) + H_\alpha(A^T) - H_\alpha(A^X, A^T)) - \gamma D_{KL}(M \| M_p)\right]. \tag{11}$$

Here, Kullback-Leibler divergence $D_{KL}(M \| M_p)$ measures the distance between distributions of M and M_p, and γ is a weight that regulates the participation of the prior information in learning the attribution M.

This extension is particularly useful when the mask prior M_p can be readily acquired using conventional methods within the domain. Moreover, by enabling users to refine M_p as necessary and incorporating it into the training, DIB-X can generate explanations tailored to users' requirements, resulting in more *interactive* explanations. The benefit of the extension will be more discussed in the Experiments section.

4. Experiments

In this section, we evaluate the performance and explainability of our proposed DIB-X method for image classification using two marine environment monitoring datasets, including multi-frequency echosounder data and images of seal pups on sea ice. Further details on these datasets are provided in Sections 4.1-4.2. We also provide a detailed account of the implementation of DIB-X, including additional information on data preparation, comparison methods, and the network architecture. These details are outlined in Sections 4.3-4.5. Our evaluation results, including visualizations, are presented in Section 4.7. As a sanity check and in order to be able to compare to alternative approaches, we also include experiments using the well-known MNIST dataset, where the implementation details and the results are presented in Appendix A.

4.1. Seal pup images on sea ice

The ice breeding harp and hooded seals are both abundant species in the North Atlantic. There are two geographically separate populations of hooded seals and three of harp seals. These have historically been exploited and managed separately. Thus, there is a need to assess the status and monitor changes in abundance in all populations to manage the respective harvests responsibly. Knowledge of seal population sizes is required to estimate the potential interaction of these species on other marine organisms, including commercially important fish species.

In a management framework, precise estimates of key parameters in population models are vital in order to provide reliable future predictions of the population. To obtain this, independent estimates of pup production using photographic or visual aerial strip transect surveys are used to determine the abundance of harp and hooded seals in the Northwest Atlantic [35], the Greenland Sea [10], and in the White Sea [36]. The total abundance is subsequently estimated by fitting a population model to the independent estimates of pup production while incorporating removals and reproductive rates [37]. The number of seal pups are counted either visually along an entire transect (with a known strip width) or from aerial images taken along a transect. A number of parallel transects are surveyed to cover an entire patch of seals. To obtain estimates of total harp or hooded seal populations several thousand of images are typically required [37].

Manual analysis of the photographs is extremely time consuming and costly, and involves subjective human interpretation by trained experts. The spatial distribution of the seals varies substantially. Typically, the ice breeding seals will cluster, but due to substantial ice drift the seals might be scattered over large areas. Often only a small fraction of the images taken contains seals, and typically 70-90 percent of harp and hooded seal images are empty.

The seal pup dataset consists of aerial photos (RGB) with corresponding annotations indicating the the position and species of all seal pups in the images. The aerial photos were acquired during surveys in the West Ice in 2007, 2012 and 2018, and Canada in 2008, 2012 and 2017. The resolution is about 2 or 3 centimeters, depending on the altitude of the aircraft. The seal pup images used in this study are manually annotated into three classes, namely harp seal, hooded seal, and background.

4.2. Multi-frequency echosounder data

Multi-frequency echosounder data reflects underwater objects by emitting varied acoustic frequencies. Acoustic target classification (ATC) aims to identify and classify marine life using this data, offering potential in fisheries management and ecosystem assessments [38]. However, ATC presents challenges due to the high variability of echosounder data and factors influencing backscattered echoes, such as size, shape, orientation, and composition [12].

Deep learning-based methods show promise in ATC, but their lack of explainability poses challenges. ATC is a multidisciplinary field, and explainability is crucial for experts' acceptance of the neural networks' decisions. Integrating diverse knowledge enhances deep learning models' reliability and robustness, addressing challenges related to echosounder data variability [39, 40, 38].

To this end, we propose integrating an attribution mask prior, denoted as M_p, into our DIB-X method to facilitate learning the attribution mask M in the multi-frequency echosounder data. The attribution mask prior M_p is obtained using the manual thresholding method, as described in [12], which sets a threshold to differentiate target species from the background in the echosounder data. Although the thresholding method's effectiveness may be limited due to the data's noisy nature, it offers ease of implementation. Once a suitable threshold is identified, the approximate location of the target species can be determined. Consequently, the knowledge within M_p serves as a foundational element for the network to learn M in DIB-X. Further details will be provided in Section 4.5.

Data	Seal pup data	Echosounder data
No. of classes	3	3
No. of training samples	3,000	10,200
No. of test samples	750	2,550
No. of validation samples	150	450
Input image size	128x128	128x128
No. of channels	3	4
Modality	RGB	Acoustic
Available domain knowledge	N/A	Threshold criterion [12]

Table 1. Additional details for the datasets used in this work.

The multi-frequency echosounder data is collected during a sandeel survey in the Norwegian North Sea. Sandeels (*Ammodytes marinus*) are small, swim bladder-less fish that primarily burrow in sandy seabeds with few silt and clay particles. The sandeel survey investigates the North Sea ecosystem to better understand the distribution, behavior, and ecology of sandeels and their relationship to other marine species [12].

Since 2005, the Norwegian Institute of Marine Research conducts acoustic trawl surveys for sandeels in the north-eastern North Sea during April and May [41]. The surveys utilize research vessels equipped with multi-frequency Simrad EK60 echosounder systems with transducers operating at 18, 38, 120, and 200 kHz, except in 2012, which uses a Simrad ME70 sonar for 120 kHz data. The echosounder systems are calibrated before each survey, and during operation, pulse duration and ping repetition frequency are set to 1.024 milliseconds and 3-4 Hz for all frequencies, respectively. Vessel speed is maintained at approximately 10 knots.

Echosounder observations for each frequency channel are recorded as frequency-specific values of the volume backscatter coefficient (s_v), representing the average backscatter intensity per cubic meter. Each s_v value corresponds to a pixel in the two-dimensional echosounder data. In the physical context, the horizontal and vertical lengths of a single s_v are set to one second and 19.2 centimeters, respectively. These values are determined by the given pulse duration and the horizontal resolution of the primary frequency channel, which is 200 kHz in this case, due to the highest sandeel signal-to-noise ratio.

4.3. Data preparation

In this study, we analyze two distinct image datasets that are highly relevant to monitoring the marine environment, including the seal pup data [10] and the multi-frequency echosounder data [11, 41]. We choose these datasets due to their unique characteristics and relevance to the field of image analysis. Table 1 provides a comprehensive overview of the datasets, including the number of classes, number of training, validation, and test samples, image sizes, and prior domain knowledge available.

To facilitate the Hadamard product with the attribution mask M ranging from zero to one, each pixel in the input X of the seal pup data is standardized to have a range from zero to one. For the multi-frequency echosounder data, each pixel is standardized to have a wider range from zero to two, which is necessary to capture the diverse patterns in the data. No data augmentation is performed for any of the datasets.

The multi-frequency echosounder data undergoes a conventional preprocessing protocol that has been used in similar studies [11]. The protocol involves converting all s_v values into a decibel (dB) unit denoted as dB re $1m^{-1}$, which represents the attenuation in decibels relative to a reference level of 1 meter traveled by the acoustic wave. Then, a minimum threshold of -75 dB re $1m^{-1}$ is set for the converted s_v values, and the minimum value is assigned to any values below the threshold or missing. Expert analysts use their domain knowledge and a post-processing software called Large Scale Survey System (LSSS) [42] to manually annotate the multi-frequency echosounder data. Each pixel in the data is assigned a class label of sandeel (SE), other fish species (OT), or background (BG) based on its characteristics.

Due to the long features along the horizontal axis resulting from several weeks of navigation, patches are extracted from the echosounder data using the image sizes specified in Table 1 to effectively train a neural network on this data. The entire echosounder data is then divided into equal bins of a given patch size with no overlap, and a class label is assigned to each of the binned patches based on the annotated pixels in them. In some cases, a patch contains two classes, which are excluded from the datasets of interest. After class labeling all the echosounder patches, the same number of patches per class are randomly sampled.

4.4. Comparison methods

To evaluate the effectiveness of our proposed DIB-X method, we select three benchmark methods for comparison. The first method is VIB-X, inspired by the recent work on variational information-bottleneck for interpretation (VIBI) [15]. VIBI is based on the IB framework [13] similar to DIB-X. However, VIBI estimates the mutual information $I(X;T)$ using a variational approach, which differs from the deterministic approach proposed in DIB-X. Specifically, VIBI derives a variational lower bound for $I(X;T)$, which is then approximated by the Kullback-Leibler divergence $D_{KL}(p(m|x)||r(m))$, where $p(m|x)$ indicates the distribution of the attribution mask M, and $r(m)$ is a prior distribution of the attribution mask M chosen in advance. However, VIBI is not intended to work as a self-explaining scheme, as the neural network in that work seeks to explain an independently obtained classifier. To make VIBI more comparable to DIB-X in terms of self-explainability, we modify VIBI in a self-explainable scheme and refer to this modified approach as VIB-X.

To generate an attribution mask for each input in VIB-X, the most relevant k sub-regions to the target are sampled using the Gumbel-softmax [31]. The vectorized output of the *explainer* is used as a basis for the Gumbel-softmax sampling. In addition, multiple masks can be generated using the same ground for each input. In this case, a final attribution mask M is obtained by aggregating based on the sub-regions with the highest activations among the multiple masks. In this study, we aggregate ten masks for each image to obtain the final attribution mask M, where each mask has four sub-regions activated, e.g., $k = 4$. The final attribution mask M creates the bottleneck representation T using the Hadamard product with the input X, and the *classifier* classifies the T to given classes.

The other two methods, Grad-CAM [19] and LIME [22], are well-known *a-posteriori* methods that rely on pretrained classifiers. Grad-CAM [19] is a gradient-based method that produces a coarse localization map highlighting important regions in the image. LIME [22] is a perturbation-based method that synthesizes neighboring image instances by occluding superpixels and fits a linear model to explain the contributions of occluded features. For LIME [22], each superpixel has a size of 8x8 to ensure the same resolution of explanation as DIB-X and VIB-X produce.

4.5. Implementation

The network architecture consists of two modules, namely the *explainer* and the *classifier*, which simulate the IB framework. More detailed information is presented in Figure B.7 in Appendix B. For DIB-X, a hyperbolic tangent layer (tanh) is added after the ReLU layer [43] at the end of the *explainer* to ensure that each pixel value in M is between zero and one. The size of M is selected based on both computational complexity and the resulting explanation resolution for the input, which is determined by the architecture of the *explainer*. In this case, nearest neighbor interpolation (x8) is applied to M to match the size of the input X.

To ensure a fair comparison, we modify the publicly available codes for the benchmark methods to use the same network as DIB-X depicted in Figure B.7. For instance, the *a-posteriori* comparison methods, such as Grad-CAM and LIME, use a pretrained *classifier* in their implementation. On the other hand, the self-explainable methods, including the proposed DIB-X and VIB-X, employ the *explainer-classifier* architecture illustrated in Figure B.7. Additional implementation details are summarized in Table 2.

DIB-X requires several hyperparameters, including the order α, the kernel width σ of Rényi's entropy, the Lagrange multiplier β associated with the *minimality* of the IB framework, and a weight γ related to the attribution mask prior M_p. First, we set α to 1.01, which is a value commonly used in the literature [34, 14] as it approximates Shannon's entropy [27].

Second, the kernel width σ controls the locality of Rényi's entropy. To ensure an accurate estimation of the entropy, it's important to avoid choosing extremely small or large values of σ [14]. When σ is small, the Gram matrix becomes more similar to the identity matrix, resulting in the eigenvalues becoming more similar to each other. On the other hand, when σ is large, the Gram matrix approaches an all-ones matrix, causing most of the eigenvalues to approach zero. Several approaches have been proposed for selecting the optimal value of σ based on the distribution of the data [44]. We apply a conventional approach based on the mean value of the k-nearest distances for each sample [34]. The value of k is carefully selected based on the classification performance of each dataset, such as $k = 6$ for the seal pup dataset and $k = 9$ for the multi-frequency echosounder data. With the selected k, the resulting value of σ is calculated as the average of the mean values for all samples.

Data	Seal pup data				Echosounder data			
Model	DIB-X	VIB-X [15]	LIME [22]	Grad-CAM [19]	DIB-X	VIB-X [15]	LIME [22]	Grad-CAM [19]
β of IB (Eq. 11)	0.02	0.1	N/A		0.005	0.1	N/A	
Batch size	128	128	128	128	128	128	128	128
No. of epochs	250	1000	500	500	500	1000	250	250
Learning rate	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0006	0.0006
Size of attribution (M)	16x16	16x16	16x16	7x7	16x16	16x16	16x16	7x7
No. of channels in M (d)	1	1	1	1	4	4	1	1
Momentum	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
Weight decay	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4
Optimizer	Stochastic Gradient Descent							

Table 2. Implementation details relevant to the reported experiment results for each setting.

Third, the Lagrange multiplier β is a crucial hyperparameter that determines the level of *minimality* of the bottleneck representation T in the IB framework. By controlling the extent to which mutual information $I(X;T)$ is incorporated into the learning process, β represents the relative influence of $I(X;T)$ on seeking *minimality* at T, while the network is trained to maximize $I(T;Y)$ with a fixed weight of one.

We carefully test a range of β values using the validation set, following the conventional grid search protocol [45] to identify the optimal configuration that achieves the best classification performance. For both datasets, we initially search for β values between zero and one with an interval of 0.1, and then further refine the search within a range of 0 to 0.1 at intervals of 0.02. For the echosounder data, we narrow down our search for the optimal value within a range of 0 to 0.02 at intervals of 0.005. The optimal β value can be found in Table 2. As an additional reference, we include the case of $\beta = 0$ to examine the performance difference when *minimality* is not pursued in DIB-X.

As illustrated in Equation 11, domain knowledge is integrated into the training through a mask prior M_p, which has binary representations based on a thresholding criterion [12]. The threshold criterion, set at -63 dB re $1m^{-1}$ for the primary frequency of 200 kHz, is applied to the dB-converted pixel values (s_v values). This results in the binary attribution mask prior M_p being defined as

$$M_p = \begin{cases} 1 & \text{for } dB(s_v) \geq -63 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

In this case, $dB(s_v)$ represents the dB-converted s_v values.

The weight γ governs the influence of domain knowledge to the objective function by regulating $D_{KL}(M||M_p)$ in Equation 11. We aim for the explanations generated by DIB-X to be based on the mask prior M_p while also fulfilling other requirements, such as *sufficiency* and *minimality*. Hence, we test several values of γ based on the previously selected β, and include in the evaluation the results for DIB-X with $\gamma = 0.005, \beta = 0.005$, which yield the highest classification performance.

4.6. Assessment of explanation quality

In this section, we evaluate the effectiveness of the attribution mask provided by the *explainer* module. To assess the quality of the explanation, we visually compare the figures from the models with the aim of assessing how well the region captured by the attribution mask matches the region in the input that is relevant to the given classes. We also investigate how the introduction of the hyperparameter β leads to a *minimal* representation of the attribution mask, enabling us to gain a deeper understanding of the most important features for classification.

Seal pup data. Figure 2 displays nine randomly selected images of test seal pup data along with their attribution masks generated by five different analysis methods, including DIB-X with $\beta = 0.02$, DIB-X with $\beta = 0$, VIB-X, LIME, and Grad-CAM. The top row shows the original seal pup images, while the remaining rows show the attribution masks M overlaid on the images, generated by each method. The attribution mask M itself is colored in orange in the figure, where each pixel in M is normalized to a value between zero and one. All methods are designed to assign higher M values to regions that are more relevant to the prediction. To optimize the legibility of the figure, the attribution masks in Figure 2 are plotted using the $M > 0.7$ criterion. The bottleneck representation T is computed as the element-wise product of the input and the attribution mask, which highlights the relevant features in the input.

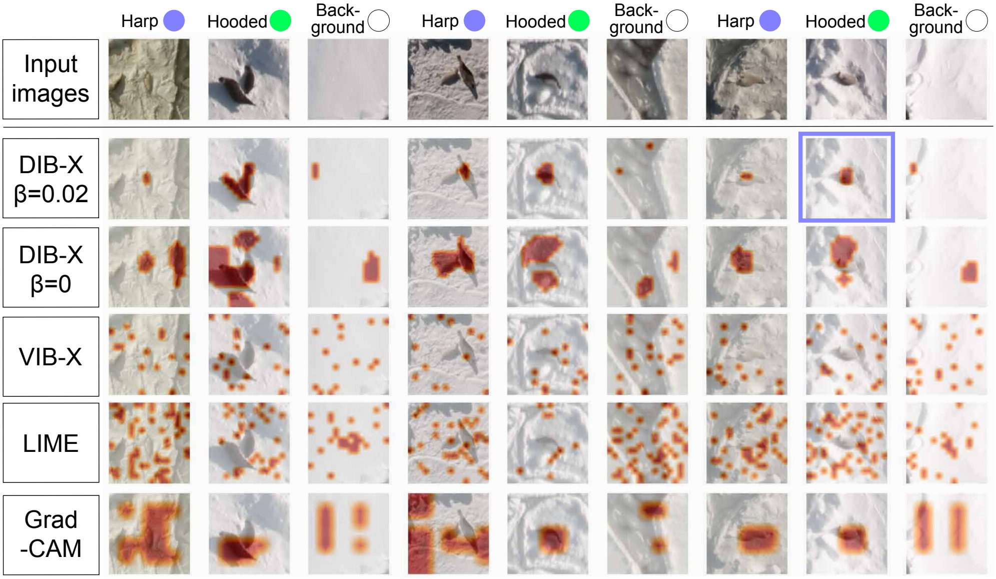


Figure 2. Randomly selected input seal pup data (top row) and the corresponding attribution masks overlaid on the images. The ground-truth class of each image is given at the top. The colored box indicates a misclassified sample by the *classifier*, where the color represents the failed prediction.

Notably, both DIB-X models, shown in both the second and the third rows of the figure, are effective at capturing the seal pups in the input images with some differences. The main difference between DIB-X with $\beta = 0.02$ and DIB-X with $\beta = 0$ is the pursuit of a *minimal* representation of the attribution mask. In the figure, the attribution mask of DIB-X with $\beta = 0.02$ (second row), which pursues *minimality*, shows a more concise representation of highlighted regions of the seals, compared to DIB-X with $\beta = 0$ (third row), which does not seek minimality.

While VIB-X appears to successfully capture some of the seals in the image, it also captures unrelated regions together, creating a scattered pattern that can be difficult for users to understand. The LIME method also produces a scattered pattern similar to VIB-X, with multiple small regions sampled across the image and some additional highlighted regions, leading to more dispersed attribution masks than VIB-X. Grad-CAM generates more focused masks compared to VIB-X and LIME. However, its explanation resolution is lower due to the the network architecture of the *classifier*, resulting in a less detailed explanation compared to other methods.

Multi-frequency echosounder data. In contrast to the seal pup data, pixel-level ground truth annotations are available for the multi-frequency echosounder data, which allow us to quantitatively evaluate the effectiveness of the explanation for each frequency channel. To do this, we first convert the attribution mask M into a pixel-level prediction of the input. This is achieved by synthesizing the mask with the class prediction from the *classifier* module. Next, we measure the predictive performance of this synthesized pixel-level prediction using the corresponding annotation, where AUROC is chosen as the metric.

Tables 3-4 present the AUROC value per frequency channel for each model and class, with the ROC curves illustrated in Figure B.8 in the Appendix B. The AUROC values in Tables 3-4 are identical. However, Table 3 emphasizes performance differences between models, while Table 4 highlights differences in the frequency-specific features of the two fish classes for each model. Our analysis does not focus on analyzing attribution masks for the BG class, as this class is considered non-informative by the attribution masks for the fish classes and is therefore discarded.

From Table 3, it is evident that DIB-X, particularly with the configuration of $\beta = 0.005$ and $\gamma = 0.005$, outperforms

AUROC		SE				OT				Average		
Freq. (kHz)		18	38	120	200	18	38	120	200	SE	OT	Total
DIB-X	$\beta = 0.005, \gamma = 0.005$	0.6135	**0.9459**	**0.9313**	**0.9511**	0.8146	**0.7670**	0.6878	0.7683	**0.8605**	0.7594	**0.8099**
	$\beta = 0.005, \gamma = 0$	0.8977	0.5981	0.8856	0.9043	**0.8313**	0.6833	**0.7943**	0.7791	0.8214	**0.7720**	0.7967
	$\beta = 0, \gamma = 0$	**0.9185**	0.5382	0.9002	0.8971	0.8200	0.4489	0.7884	**0.7970**	0.8135	0.7136	0.7635
VIB-X [15]		0.8329	0.8286	0.8304	0.8302	0.6171	0.6241	0.6198	0.6171	0.8305	0.6195	0.7250
LIME [22]		0.7668	0.8579	0.8671	0.8688	0.7585	0.6788	0.6455	0.7257	0.8402	0.7021	0.7711
Grad-CAM [19]		0.8578				0.7112				0.8578	0.7112	0.7845

Table 3. AUROC values of the explanation for two fish classes, e.g., sandeel (SE) and other species (OT).

Class	Freq. (kHz)	DIB-X			VIB-X [15]	LIME [22]	Grad-CAM [19]
		$\beta = 0.005$ $\gamma = 0.005$	$\beta = 0.005$ $\gamma = 0$	$\beta = 0$ $\gamma = 0$			
SE	18	0.6135	0.8977	**0.9185**	**0.8329**	0.7668	0.8578
	38	0.9459	0.5981	0.5382	0.8286	0.8579	
	120	0.9313	0.8856	0.9002	0.8304	0.8671	
	200	**0.9511**	**0.9043**	0.8971	0.8302	**0.8688**	
OT	18	**0.8146**	**0.8313**	**0.8200**	0.6171	**0.7585**	0.7112
	38	0.7670	0.6833	0.4489	**0.6241**	0.6788	
	120	0.6878	0.7943	0.7884	0.6198	0.6455	
	200	0.7683	0.7791	0.7970	0.6171	0.7257	

Table 4. Per class performance comparison with respect to AUROC value and accuracy of the multi-channel echosounder data

the other models across the majority of frequency channels and fish classes. DIB-X with $\beta = 0.005$ and $\gamma = 0.005$ achieves the highest total average (0.8099), as well as the highest average value for the SE class (0.8605). This configuration excels, particularly in the 38, 120, and 200 kHz frequency channels for the SE class and the 38 kHz frequency for the OT class. In addition, DIB-X with $\beta = 0.005$ and $\gamma = 0$ slightly underperforms compared to DIB-X with $\beta = 0.005$ and $\gamma = 0.005$, while achieving the highest values for several cases, including the average value for the OT class (0.7720), and 18 and 120 kHz frequency channels for the OT class.

VIB-X shows relatively consistent performance in the SE class, with an average of 0.8305, but demonstrates weaker performance in the OT class, with an average of 0.6195. LIME displays slightly better performance than VIB-X, with an average of 0.8402 for the SE class, 0.7021 for the OT class, and a total average of 0.7711. Grad-CAM does not provide frequency-wise values due to its nature of generating explanations, but achieves a relatively high value of 0.8578 for the SE class. However, these comparison models, including Grad-CAM, still achieve lower total average values than the DIB-X configurations, such as $\beta = 0.005, \gamma = 0.005$ and $\beta = 0.005, \gamma = 0$.

It is evident from our analysis of Table 3 that integrating domain knowledge into the model helps generate attribution masks that better match pixel-level annotations. This is particularly notable when comparing the performance of DIB-X with $\beta = 0.005$ and $\gamma = 0.005$ to that of DIB-X with $\beta = 0.005$ and $\gamma = 0$, where γ is a hyperparameter controlling the incorporation of the domain knowledge through the mask prior M_p.

Furthermore, the domain knowledge used in this study is based on threshold criteria that facilitate the discrimination of the SE class from the background during manual annotation of the echosounder data [12]. Our analysis indicates that the difference in AUROC values between the SE and OT classes suggests the effectiveness of domain knowledge, particularly for the SE class. For instance, DIB-X with $\beta = 0.005$ and $\gamma = 0.005$ performs better for the SE class on the majority of the frequency channels (38, 120, 200 kHz) compared to other configurations, while this is not the case for the OT class.

Table 4 compares the performance of the SE and OT classes at different frequency channels in terms of AUROC values. We observe that the SE class achieves highest AUROC values at 200 kHz for two proposed DIB-X models, including $\beta = 0.005, \gamma = 0.005$ and $\beta = 0.005, \gamma = 0$, and LIME, while the OT class achieves higher AUROC values at lower frequency channels for all models. This observation aligns with the conventional understanding that the SE species achieves the best signal-to-noise ratio at 200 kHz [12]. As a result, the primary frequency for the sandeel survey is typically set to 200 kHz [41].

The visual evaluations shown in Figures 3-4 support the quantitative analysis results presented in this study. Figure 3 provides a comparison of attribution masks generated by multiple models across four frequency channels. As

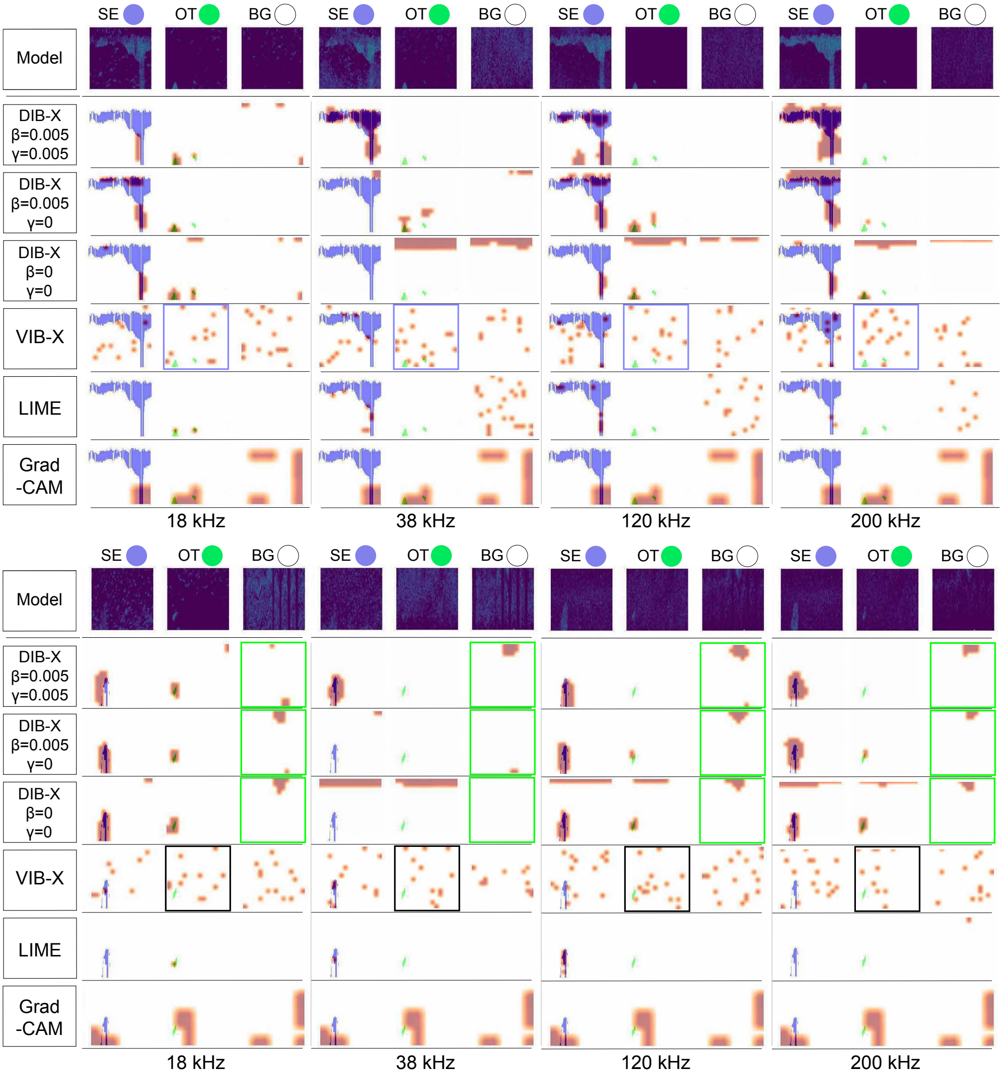


Figure 3. Comparison of attribution masks generated by multiple models across four frequency channels of the multi-frequency echosounder data.

demonstrated in Tables 3-4, the DIB-X models with $\beta = 0.005$ effectively capture potential fish schools. Notably, the SE class (blue) is captured more frequently at higher frequency channels (e.g., 200 kHz), while the OT class (green) is captured more frequently at lower frequency channels (e.g., 18 kHz).

Figure 4 shows the mask prior M_p. The available domain knowledge, based on the primary frequency of the sandeel survey (200 kHz) due to the best noise-to-ratio, leads us to establish a threshold criterion for the sandeel fish schools, e.g., the SE class. We observe that the mask prior provides more identifiable information at this frequency

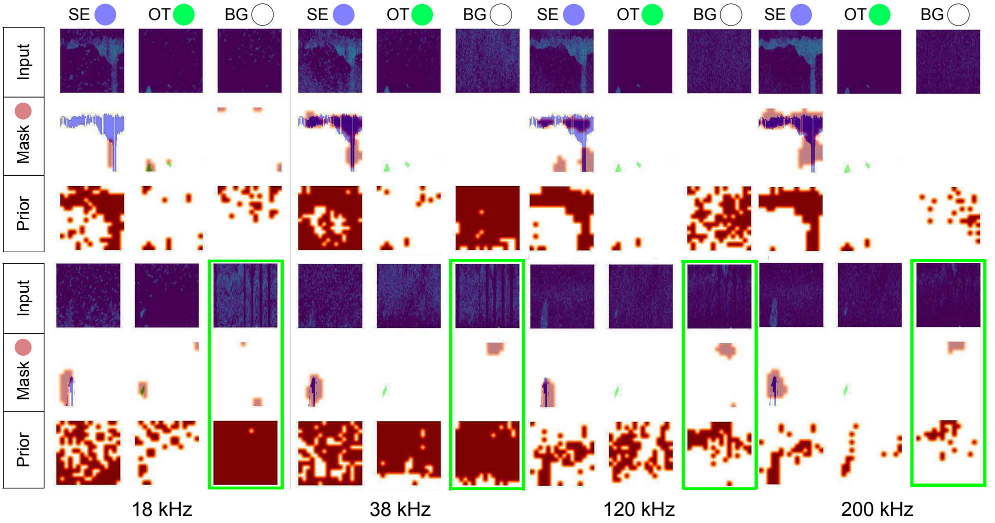


Figure 4. Visualization of the multi-frequency echosounder data and corresponding attribution masks M and mask priors M_p per frequency channel.

channel, compared to other frequency channels. Additional figures in Appendix Appendix B, including Figures B.9-B.13, further support the robustness of our proposed DIB-X.

4.7. *Evaluation of predictive performance*

We present the average and per-class image classification performance on the test dataset of the different methods for two marine environment monitoring datasets. Tables 5-6 show the results from the seal pup data. Tables 7-8 show the results from the multi-frequency echosounder data. Four metrics, namely AUROC, Accuracy, Cohen-Kappa, and F1 (macro), are utilized to evaluate the different methods. In the tables, each bold value in the tables denotes the largest value among the methods for that specific evaluation metric.

To investigate the impact of the hyperparameter β on predictive performance, we test the case of $\beta = 0$ for DIB-X. In this case, the network is trained using an objective function that maximizes $I(T;Y)$ to ensure that the bottleneck representation T contains *sufficient* information for prediction, without taking into account the trade-off between *sufficiency* and *minimality*. However, the case of DIB-X with $\beta = 0$ still differs from the *a-posteriori* methods, such as LIME and Grad-CAM. This is because DIB-X with $\beta = 0$ uses the *explainer-classifier* architecture, while the *a-posteriori* methods rely solely on the *classifier* module.

Seal pup data. Table 5 provides a comparison of the average performance of the models on the seal pup data. Our proposed DIB-X with $\beta = 0.02$ achieves the best results across all metrics, demonstrating its effectiveness in the image classification task. Table 6 displays the classification performance for each class. Consistent with the results in Table 5, DIB-X with $\beta = 0.02$ exhibits better performance in AUROC across all three classes, including harp seal pup (0.9861), hooded seal pup (0.9646), and background (BG, 0.9924), and achieves the highest accuracy for the hooded seal pup class (0.7857), as well as the top F1 scores for both hooded seal pup (0.8635) and background (0.9558) classes.

Our analysis reveals a few notable findings. Firstly, our proposed DIB-X outperforms VIB-X, indicating that the deterministic measure of mutual information $I(X;T)$ contributes more to the improved classification performance than the variational measure used in VIB-X. Secondly, we observe that when comparing two DIB-X setups with different values of the hyperparameter β, e.g., $\beta = 0.02$ and $\beta = 0$, pursuing *minimal* representation at the bottleneck T, i.e., DIB-X with $\beta = 0.02$, can improve predictive performance. This finding suggests that pursuing *minimality* can act

Table 5. Average performance comparison of the seal pup data

Model	AUROC	Accuracy	Cohen-Kappa	F1 (macro)
DIB-X ($\beta = 0.02$)	**0.9811**	**0.9000**	**0.8495**	**0.8974**
DIB-X ($\beta = 0$)	0.9757	0.8922	0.8371	0.8890
VIB-X [15]	0.9661	0.8781	0.8154	0.8737
LIME [22], Grad-CAM [19]	0.9728	0.8719	0.8068	0.8678

Model	AUROC			Accuracy			F1 score		
	Harp	Hooded	BG	Harp	Hooded	BG	Harp	Hooded	BG
DIB-X ($\beta = 0.02$)	**0.9861**	**0.9646**	**0.9924**	0.9528	**0.7857**	0.9587	0.8730	**0.8635**	**0.9558**
DIB-X ($\beta = 0$)	0.9812	0.9565	0.9894	**0.9817**	0.7251	0.9667	**0.8882**	0.8369	0.9418
VIB-X [15]	0.9627	0.9465	0.9892	0.9500	0.7122	0.9628	0.8507	0.8174	0.9532
LIME [22], Grad-CAM [19]	0.9736	0.9569	0.9879	0.9722	0.6729	**0.9714**	0.8617	0.7998	0.9419

Table 6. Per class performance comparison with respect to AUROC value, accuracy, and f1-score of the seal pup data

as a regularizer in training the network, encouraging the network to avoid learning non-informative features, such as redundancy, to achieve learning concise yet comprehensive feature representations relevant to the label space [15].

Lastly, we also observe that DIB-X with $\beta = 0$ outperforms *a-posteriori* methods such as LIME and Grad-CAM in terms of prediction performance. Given that the methods mentioned above use the same objective function of minimizing the cross-entropy, this result suggests that the *explainer* module of DIB-X to some extent acts as the *classifier*. In other words, the *explainer* learns discriminative features, providing additional learning capacity to the *classifier* rather than explaining it. The linear combination between the *explainer* and the *classifier*, such as the Hadamard product at the bottleneck, enables the backpropagation of the gradient computed from the output of the *classifier*, facilitating this extended functionality of the *explainer* module.

To ensure a rigorous comparison of DIB-X with *a-posteriori* methods, we argue that it is necessary to restrict the *explainer* module from providing additional learning capacity to the *classifier*. In other words, each module in the sequentially connected network should learn a fundamentally distinct feature representation based on its intended purpose.

To achieve this, one possible approach is to implement additional constraints at the bottleneck through the attribution mask M. These constraints should ensure that each module can learn a purpose-oriented representation without interfering with the capacity of the other module. Further exploration is required to clarify the specifics of these constraints, which falls beyond the scope of this paper.

Multi-frequency echosounder data. Table 7 provides a comparison of the average classification accuracy of five different models on multi-frequency echosounder data. Table 8 displays the classification performance for each class, and the obtained results are consistent with those in Table 7. The best overall performance across all metrics is achieved by DIB-X with $\beta = 0.005$ and $\gamma = 0.005$, outperforming the comparison models for the majority of frequency channels and classes, as shown in Table 8. For models that do not incorporate the mask prior M_p (i.e., cases with $\gamma = 0$), DIB-X with $\beta = 0.005$ and $\gamma = 0$ surpasses other methods, attaining the highest values for AUROC (0.9371), Accuracy (0.8603), and F1 score (0.8233) in the BG class.

Our analysis of the multi-frequency echosounder data has led to several findings. Firstly, the findings we have obtained from the seal pup data are also applicable to the multi-frequency echosounder data. The findings include the advantage of the deterministic mutual information measure $I(X;T)$ over the variational measure applied to VIB-X, as well as the importance of pursuing *minimality* to prevent the network from learning redundant features. They also include that the *explainer* module should be constrained to learn proper feature representations that explain the *classifier*, rather than simply aiding the classification task.

Secondly, comparing two cases of DIB-X leveraging the mask prior M_p, specifically $\beta = 0.005, \gamma = 0.005$ and $\beta = 0.005, \gamma = 0$, we observe that DIB-X with mask prior improves its classification performance. Based on this result, we argue that incorporating domain knowledge during training can lead to the estimation of a more grounded attribution mask by the network.

Table 7. Average performance comparison of the multi-channel echosounder data

Model	AUROC	Accuracy	Cohen-Kappa	F1 (macro)
DIB-X (β = 0.005, γ = 0.005)	**0.9237**	**0.8063**	**0.7081**	**0.8054**
DIB-X (β = 0.005, γ = 0)	0.9164	0.7874	0.6800	0.7862
DIB-X (β = 0, γ = 0)	0.9134	0.7862	0.6772	0.7836
VIB-X [15]	0.7997	0.6315	0.4463	0.6210
LIME [22], Grad-CAM [19]	0.9031	0.7590	0.6378	0.7581

Model	AUROC			Accuracy			F1 score		
	SE	OT	BG	SE	OT	BG	SE	OT	BG
DIB-X (β = 0.005, γ = 0.005)	**0.9411**	**0.8961**	0.9340	**0.8030**	**0.7716**	0.8444	**0.8402**	**0.7574**	0.8187
DIB-X (β = 0.005, γ = 0)	0.9241	0.8880	**0.9371**	0.7391	0.7611	**0.8603**	0.7908	0.7444	**0.8233**
DIB-X (β = 0, γ = 0)	0.9276	0.8782	0.9345	0.7644	0.7372	0.8563	0.8068	0.7312	0.8129
VIB-X [15]	0.7896	0.7606	0.8489	0.5275	0.5111	0.8593	0.5975	0.5691	0.6963
LIME [22], Grad-CAM [19]	0.9137	0.8743	0.9212	0.7226	0.7099	0.8446	0.7726	0.7150	0.7867

Table 8. Per class performance comparison with respect to AUROC value and accuracy of the multi-channel echosounder data

Lastly, VIB-X does not perform well on the multi-frequency echosounder data, which is in contrast to its performance on seal pup data. We suggest that this is due to the type of explanation required for multi-frequency echosounder data, which should capture the differences in frequency-specific patterns among fish species in order to provide insight aligned with the user's perspective. To address this, we generate multi-frequency attribution masks that allow us to analyze the backscattered patterns across all frequency channels for the same fish school. Notably, the masks should be acquired in a higher-dimensional space than the seal pup data, given the high-dimensional nature of the multi-frequency echosounder data. This complexity may pose diffculties for VIB-X to sample a good attribution mask using Gumbel-softmax [31], which can cause its weak performance. Moreover, [34] states that the lowerbound of variational IB approaches, including VIB-X, may not work well in practice due to challenges in ensuring the tightness of the derived lower bound.

5. Conclusion

In conclusion, we propose DIB-X, a a generic and self-explainable deep learning method that generates *minimal*, *sufficient*, and *interactive* explanations for the decisions of the model. By utilizing the information bottleneck (IB) framework [13], DIB-X ensures minimal and sufficient explanations [15]. One of the key novelties of DIB-X is the incorporation of an innovative mutual information measure, the matrix-based Rényi's α-order entropy functional [14], into the IB framework. This allows DIB-X to circumvent the variational approximation and distributional assumption typically necessitated by conventional IB-based approaches [15, 29]. Another noteworthy aspect of DIB-X is its capacity to produce explanations that align with well-established domain understanding, rendering them more *interactive*. This is accomplished by incorporating domain knowledge directly into the learning process, where the knowledge is transformed into prior information, thereby informing the explanations generated.

We evaluate DIB-X using two marine environment monitoring datasets with distinct modalities. Empirical results demonstrate that our method provides enhanced explainability compared to benchmark explainability methods [22, 19, 15], while also offering improved classification performance as an added benefit. These results validate the effectiveness of DIB-X.

Despite its merits, our method has certain limitations and offers avenues for further enhancement. From a methodological viewpoint, it is evident that further investigation into the constraints at the bottleneck is necessary to ensure that each module within the sequentially connected network learns a purpose-oriented representation without adversely affecting the capacity of other modules. Additionally, to generalize the *interactivity* principle, further robustness examination is necessary, along with methodological expansions to accommodate other representations of prior knowledge. From a practical perspective, future research should explore the application of DIB-X across diverse domains and contexts to determine its wider applicability and potential utility in various fields.

Acknowledgements

This work was financially supported by the Research Council of Norway (RCN), grant no. 270966 "Ubiquitous Cognitive Computer Vision for Marine Services" (COGMAR). The work was further financed by the RCN grant no. 309439 Centre for Research-based Innovation "Visual Intelligence" and consortium partners, as well as RCN grant no. 309512 Centre for Research-based Innovation in Marine Acoustic Abundance Estimation and Backscatter Classification (CRIMAC).

References

[1] C. M. Duarte, S. Agusti, E. Barbier, G. L. Britten, J. C. Castilla, J.-P. Gattuso, R. W. Fulweiler, T. P. Hughes, N. Knowlton, C. E. Lovelock, et al., Rebuilding marine life, Nat. 580 (7801) (2020) 39–51.

[2] Y. Cong, B. Fan, D. Hou, H. Fan, K. Liu, J. Luo, Novel event analysis for human-machine collaborative underwater exploration, Pattern Recognit. 96 (2019) 106967.

[3] C. Li, S. Anwar, F. Porikli, Underwater scene prior inspired deep underwater image and video enhancement, Pattern Recognit. 98 (2020) 107038.

[4] W. Samek, Explainable deep learning: concepts, methods, and new developments, in: J. Benois-Pineau, R. Bourqui, D. Petkovic, G. Quénot (Eds.), Explainable Deep Learning AI, Academic Press, New York, 2023, pp. 7–33.

[5] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (5) (2019) 206–215.

[6] S. Gautam, A. Boubekki, S. Hansen, S. Salahuddin, R. Jenssen, M. Höhne, M. Kampffmeyer, ProtoVAE: A trustworthy self-explainable prototypical variational model, in: Proc. Neural Inf. Process. Syst. (NeurIPS), Vol. 35, Curran Associates, Inc., 2022, pp. 17940–17952.

[7] Z. C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery., ACM Queue 16 (3) (2018) 31–57.

[8] A. Ordonez, L. Eikvil, A.-B. Salberg, A. Harbitz, S. M. Murray, M. C. Kampffmeyer, Explaining decisions of deep neural networks used for fish age prediction, PloS ONE 15 (6) (2020) e0235013.

[9] K. Malde, N. O. Handegard, L. Eikvil, A.-B. Salberg, Machine intelligence and the data-driven future of marine science, ICES J. Mar. Sci. 77 (4) (2019) 1274–1285.

[10] M. Biuw, T. A. Øigård, K. T. Nilssen, G. Stenson, L. Lindblom, M. Poltermann, M. Kristiansen, T. Haug, Recent harp and hooded seal pup production estimates in the greenland sea suggest ecology-driven declines, NAMMCO Sci. Publ. 12 (2022) 1–15.

[11] C. Choi, M. Kampffmeyer, N. O. Handegard, A.-B. Salberg, R. Jenssen, Deep semisupervised semantic segmentation in multifrequency echosounder data, IEEE J. Ocean. Eng. Preprint (2023) 1–17. doi:10.1109/JOE.2022.3226214.

[12] E. Johnsen, R. Pedersen, E. Ona, Size-dependent frequency response of sandeel schools, ICES J. Mar. Sci. 66 (6) (2009) 1100–1105.

[13] N. Tishby, N. Zaslavsky, Deep learning and the information bottleneck principle, in: Proc. IEEE Inf. Theory Workshop, IEEE, 2015, pp. 1–5. doi:10.1109/ITW.2015.7133169.

[14] S. Yu, L. G. S. Giraldo, R. Jenssen, J. C. Principe, Multivariate extension of matrix-based rényi's α-order entropy functional, IEEE Trans. Pattern Anal. Mach. Intell. 42 (11) (2019) 2960–2966.

[15] S. Bang, P. Xie, H. Lee, W. Wu, E. Xing, Explaining a black-box by using a deep variational information bottleneck approach, in: Proc. Conf. Artif. Intell. (AAAI), Vol. 35, 2021, pp. 11396–11404.

[16] K. Schulz, L. Sixt, F. Tombari, T. Landgraf, Restricting the flow: Information bottlenecks for attribution, in: Proc. Int. Conf. Learn. Representations (ICLR), 2020, pp. 1–18.

[17] D. Alvarez Melis, T. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: Proc. Neural Inf. Process. Syst. (NeurIPS), Vol. 31, Curran Associates, Inc., 2018, pp. 1–10.

[18] I. Covert, S. M. Lundberg, S.-I. Lee, Explaining by removing: A unified framework for model explanation., J. Mach. Learn. Res. 22 (1) (2021) 9477–9566.

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: Proc. Int. Conf. Comput. Vis. (ICCV), 2017, pp. 618–626.

[20] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PloS ONE 10 (7) (2015) e0130140.

[21] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: Proc. Int. Conf. Mach. Learn. (ICML), PMLR, 2017, pp. 3145–3153.

[22] M. T. Ribeiro, S. Singh, C. Guestrin, " why should i trust you?" explaining the predictions of any classifier, in: Proc. ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD), 2016, pp. 1135–1144.

[23] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proc. Eur. Conf. Comput. Vis. (ECCV), Springer, 2014, pp. 818–833.

[24] V. Petsiuk, A. Das, K. Saenko, RISE: randomized input sampling for explanation of black-box models, in: Proc. Br. Mach. Vis. Conf. (BMVC), BMVA Press, 2018, pp. 1–17.

[25] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining explanations: An overview of interpretability of machine learning, in: IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA), IEEE, 2018, pp. 80–89.

[26] A. Rényi, On measures of entropy and information, in: Proc. Berkeley Symp. Math. Stat. Probab., Vol. 1, Berkeley, California, USA, 1961, pp. 547–561.

[27] C. E. Shannon, A mathematical theory of communication, ACM Mob. Comput. Commun. Rev. (SIGMOBILE) 5 (1) (2001) 3–55.

[28] A. A. Alemi, I. Fischer, J. V. Dillon, K. Murphy, Deep variational information bottleneck, in: Proc. Int. Conf. Learn. Representations (ICLR), 2017, pp. 1–19.

[29] A. Zhmoginov, I. Fischer, M. Sandler, Information-bottleneck approach to salient region discovery, in: Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discov. Databases (ECML PKDD), Springer, 2021, pp. 531–546.

[30] X. Yu, S. Yu, J. C. Príncipe, Deep deterministic information bottleneck with matrix-based entropy functional, in: Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), IEEE, 2021, pp. 3160–3164.

[31] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, in: Proc. Int. Conf. Learn. Representations (ICLR), 2017, pp. 1–13.

[32] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, D. Hjelm, Mutual information neural estimation, in: Proc. Int. Conf. Mach. Learn. (ICML), PMLR, 2018, pp. 531–540.

[33] A. Zaidi, I. Estella-Aguerri, S. Shamai (Shitz), On the information bottleneck problems: Models, connections, applications and information theoretic views, Entropy 22 (2) (2020) e22020151.

[34] Q. Zhang, S. Yu, J. Xin, B. Chen, Multi-view information bottleneck without variational approximation, in: Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), IEEE, 2022, pp. 4318–4322.

[35] G. Stenson, L.-P. Rivest, M. Hammill, J. Gosselin, Estimating pup production of harp seals, pagophilus groenlandicus, in the northwest atlantic, Mar. Mamm. Sci. 19 (1) (2003) 141–160.

[36] V. Potelov, A. Golikov, V. Bondarev, Estimated pup production of harp seals pagophilus groenlandicus in the white sea, russia, in 2000, ICES J. Mar. Sci. 60 (5) (2003) 1012–1017.

[37] M. O. Hammill, C. den Heyer, W. Bowen, Grey seal population trends in canadian waters, 1960-2014, DFO Can. Sci. Advis. Sec. Res. Doc. (2014) 1–44.

[38] R. J. Korneliussen, Acoustic target classification, Coop. Res. Rep. - Int. Counc. Explor. Seadoi:10.17895/ices.pub.4567.

[39] N. O. Handegard, D. Tjøstheim, The sampling volume of trawl and acoustics: estimating availability probabilities from observations of tracked individual fish, Can. J. Fish. Aquat. Sci. 66 (3) (2009) 425–437.

[40] E. Ona, An expanded target-strength relationship for herring, ICES J. Mar. Sci. 60 (3) (2003) 493–499.

[41] E. Johnsen, G. Rieucau, E. Ona, G. Skaret, Collective structures anchor massive schools of lesser sandeel to the seabed, increasing vulnerability to fishery, Mar. Ecol.: Prog. Ser. 573 (2017) 229–236.

[42] R. Korneliussen, E. Ona, I. Eliassen, Y. Heggelund, R. Patel, O. Godø, C. Giertsen, D. Patel, E. Nornes, T. Bekkvik, et al., The large scale survey system-LSSS, in: Proc. Scand. Symp. Phys. Acoust., 2006, pp. 1–6.

[43] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proc. Int. Conf. Mach. Learn. (ICML), 2010, p. 807–814.

[44] J. N. Myhre, K. Ø. Mikalsen, S. Løkse, R. Jenssen, Robust clustering using a knn mode seeking ensemble, Pattern Recognit. 76 (2018) 491–505.

[45] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.

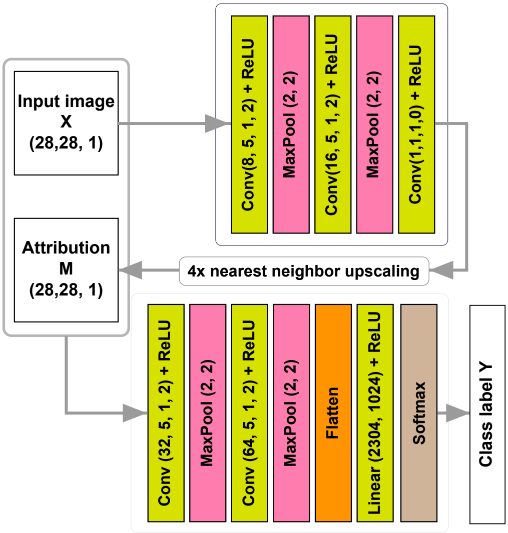


Figure .5. Proposed network architecture for MNIST dataset.

Appendix A. Evaluation on MNIST data

Although our focus in this section is primarily on marine environment monitoring datasets, the implementation details and the results from the MNIST dataset are provided in the Appendix A. For the MNIST dataset, we utilize the default split for training and testing, and no additional preprocessing or data manipulation is conducted for any of the datasets.

Appendix A.1. Network architecture

The network architecture for MNIST data is shown in Figure .5 in the Appendix A. Table A.9 presents the implementation details that are applied to MNIST dataset.

Data	MNIST			
Model	DIB-X	VIB-X [15]	LIME [22]	Grad-CAM [19]
β of IB (Eq. 11)	0.02	0.1	N/A	
Batch size	128	128	128	128
No. of epochs	100	1000	100	100
Learning rate	0.0003	0.0003	0.0003	0.0003
Size of attribution (M)	16x16	16x16	16x16	7x7
No. of channels in M (d)	1	1	1	1
Momentum	0.9	0.9	0.9	0.9
Weight decay	5e-4	5e-4	5e-4	5e-4
Optimizer	Stochastic Gradient Descent			

Table A.9. Implementation details for the MNIST dataset.

Appendix A.2. Results

Table A.10 compares the performance of four models on the MNIST dataset. The models are evaluated on four different metrics: AUROC, accuracy, Cohen-Kappa, and F1 score (macro). DIB-X with $\beta = 0.02$ outperforms all

Model	AUROC	Accuracy	Cohen-Kappa	F1 (macro)
DIB-X (β = 0.02)	**0.9998**	**0.9862**	**0.9845**	**0.9859**
DIB-X (β = 0)	**0.9998**	0.9828	0.9807	0.9816
VIB-X [15]	0.9992	0.9674	0.9635	0.9656
LIME [22], Grad-CAM [19]	0.9996	0.9801	0.9777	0.9793

Table A.10. Average performance comparison: MNIST

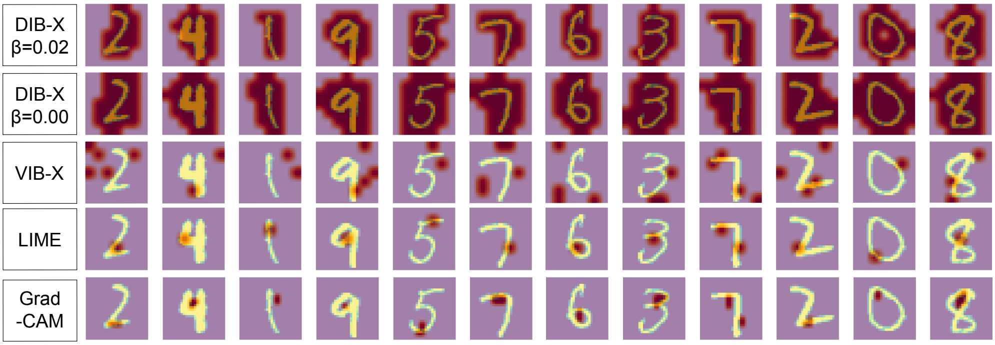


Figure A.6. Comparison of attribution masks generated by different methods on twelve randomly selected images from the MNIST test dataset. Each mask highlights the relevant region with respect to the class prediction.

other models on all four metrics. Specifically, it achieves an AUROC of 0.9998, an accuracy of 0.9862, a Cohen-Kappa score of 0.9845, and an F1 score (macro) of 0.9859. DIB-X with $\beta = 0$ also performs well, achieving the same AUROC as the top-performing model but slightly lower scores on the other three metrics.

VIB-X achieves lower scores than the DIB-X models but still performs well, achieving an AUROC of 0.9992, an accuracy of 0.9674, a Cohen-Kappa score of 0.9635, and an F1 score (macro) of 0.9656. LIME and Grad-CAM achieve the lowest scores among the four models, with an AUROC of 0.9996, an accuracy of 0.9801, a Cohen-Kappa score of 0.9777, and an F1 score (macro) of 0.9793.

Figure A.6 compares the attribution masks generated by different methods on twelve correctly classified images from the MNIST test dataset. The figure suggests that the attribution masks from DIB-X with $\beta = 0.02$ efficiently convey the most important features for classification, as they highlight the most relevant regions (i.e., numbers) compared to the other methods, including VIB-X, LIME, and Grad-CAM. Moreover, the attribution masks from DIB-X with $\beta = 0.02$ achieve this with the least number of activated pixels, making the masks more concise and visually interpretable. This indicates that the DIB-X model with $\beta = 0.02$ is better at identifying the most relevant and informative features for classification than the other methods tested, including the DIB-X model with $\beta = 0$.

Appendix B. Additional figures of the multi-frequency echosounder data

Appendix B.1. Network architecture

Figure B.7 shows the network architecture used for two marine environment monitoring datasets. In the figure, the depth of the input image, denoted as D, and the depth of the attribution, denoted as d, depend on the dataset. For the multi-frequency echosounder data, $D = d = 4$, while for the seal pup data, $D = 3$ and $d = 1$.

Appendix B.2. Comparison of ROC curves

Figure B.8 presents 24 ROC curves comparing the performance of six different explainability methods on four different frequency channels, where each subfigure has ROC curves of two different fish classes (sandeel and other species) and their average (macro). The results show that DIB-X with $\beta = 0.005, \gamma = 0.005$ achieves the highest

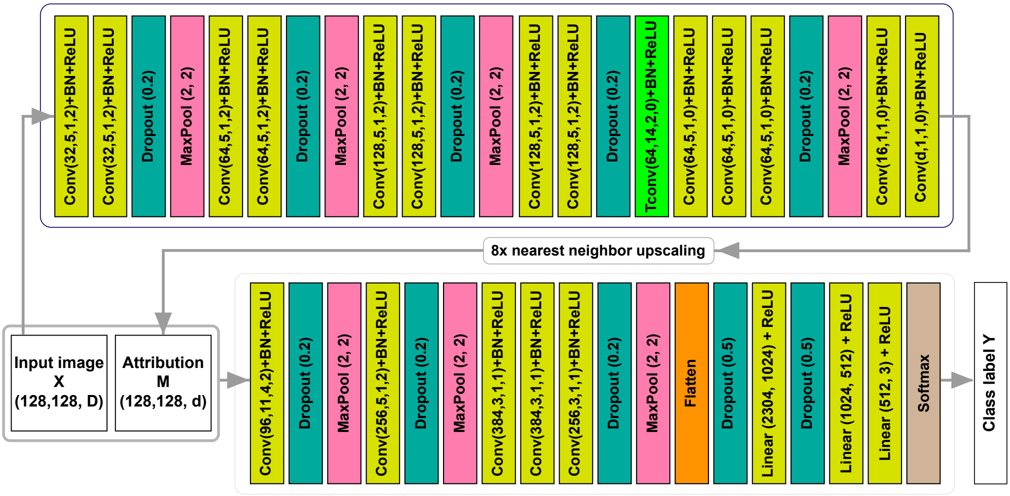


Figure B.7. Proposed network architecture for the multi-frequency echosounder data and the seal image data. The upper module shows the *explainer*, and the lower module shows the *classifier*.

TPR for all four frequency channels and for the sandeel and other species classes, as well as their average. The corresponding AUROC values are presented in Tables 3-4.

Appendix B.3. Comparison of the explanation regarding each frequency channel

Figures B.9-B.12 provide additional comparison of the attribution masks for different models, where the corresponding mask prior is presented in Figure B.13.

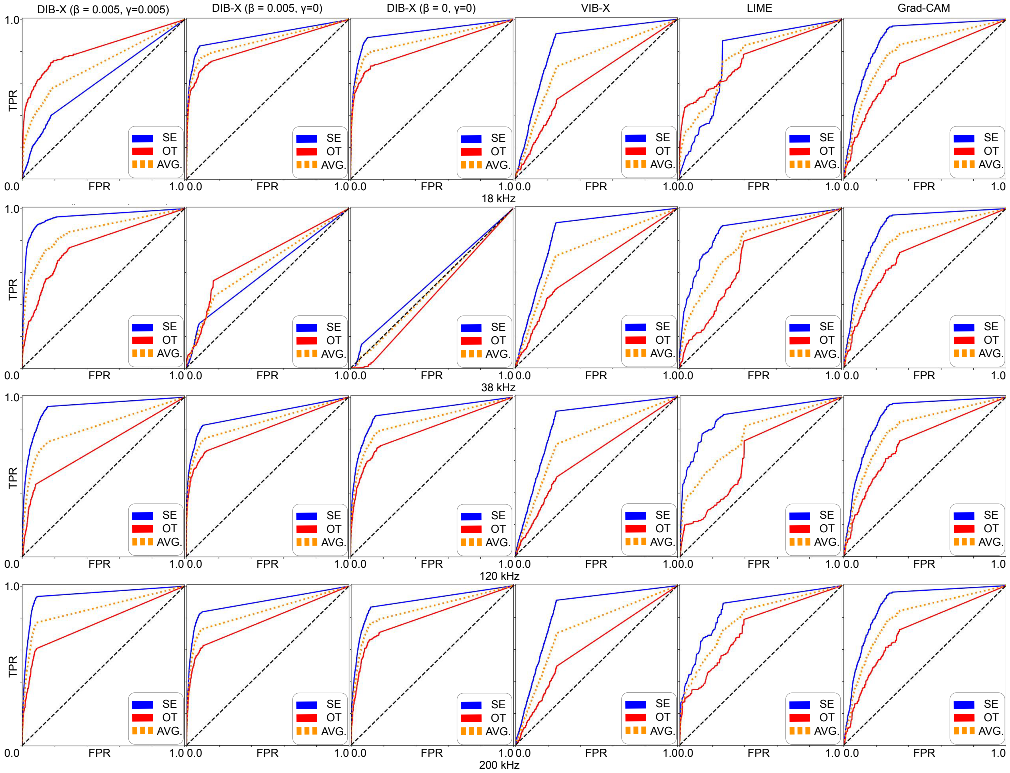


Figure B.8. This figure compares the receiver operating characteristic (ROC) curves of six different explainability methods, including DIB-X with different values of β and γ, VIB-X, LIME, and Grad-CAM, on four different frequency channels (18, 38, 120, 200 kHz). The true positive rate (TPR) is plotted against the false positive rate (FPR) for each method and frequency channel combination. Each subfigure includes three ROC curves, two for the fish classes, e.g., sandeel (SE) and other species (OT), and the average (macro).

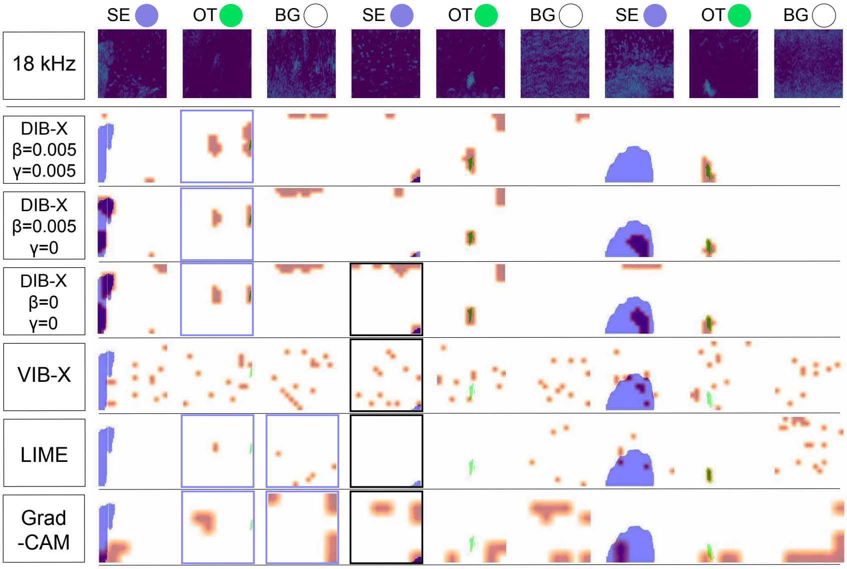


Figure B.9. Comparison of attribution masks for 18kHz of the multi-frequency echosounder data.

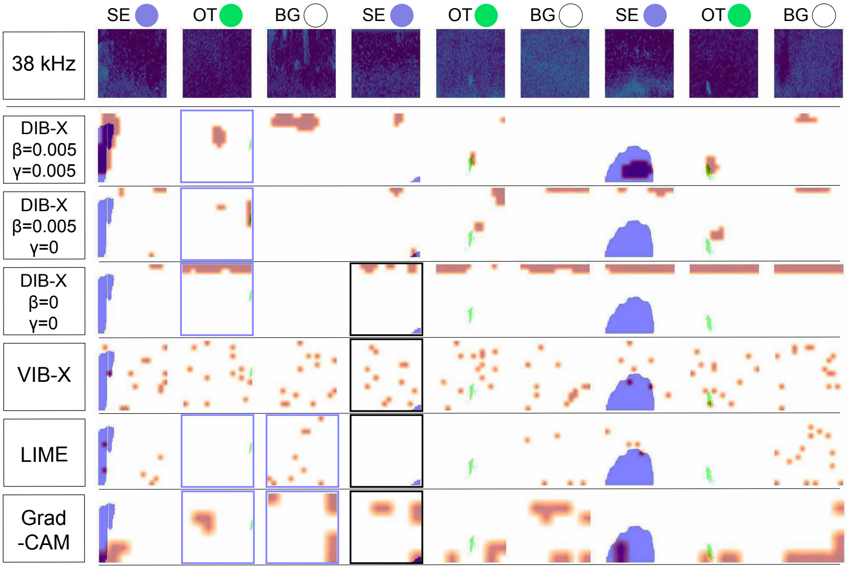


Figure B.10. Comparison of attribution masks for 38kHz of the multi-frequency echosounder data.

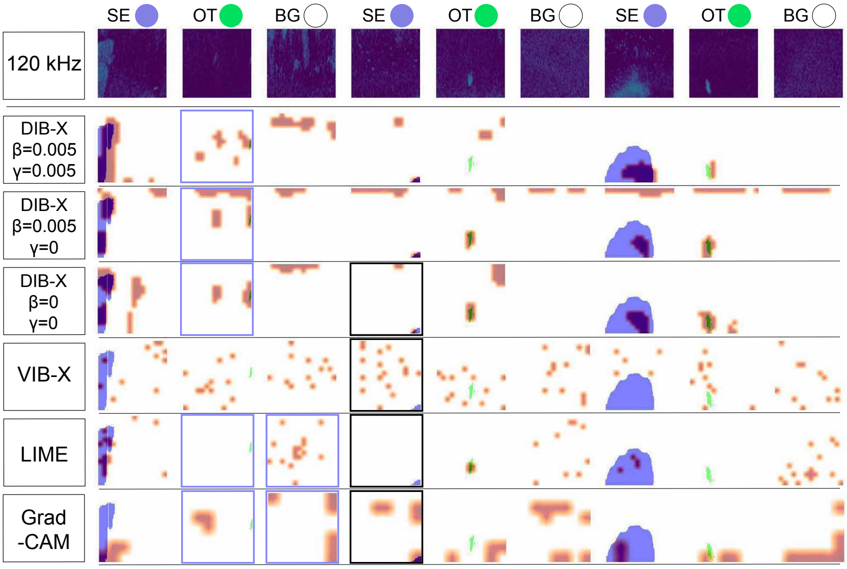


Figure B.11. Comparison of attribution masks for 120kHz of the multi-frequency echosounder data.

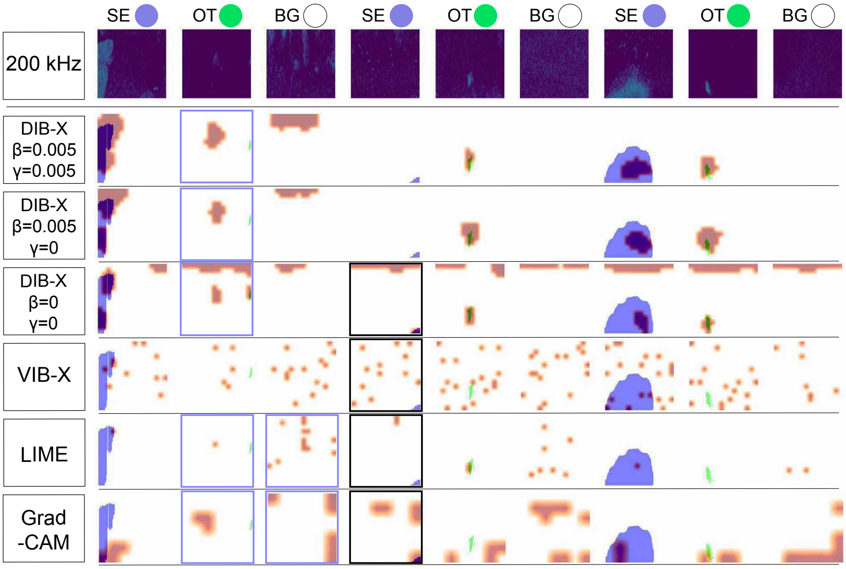


Figure B.12. Comparison of attribution masks for 200kHz of the multi-frequency echosounder data.

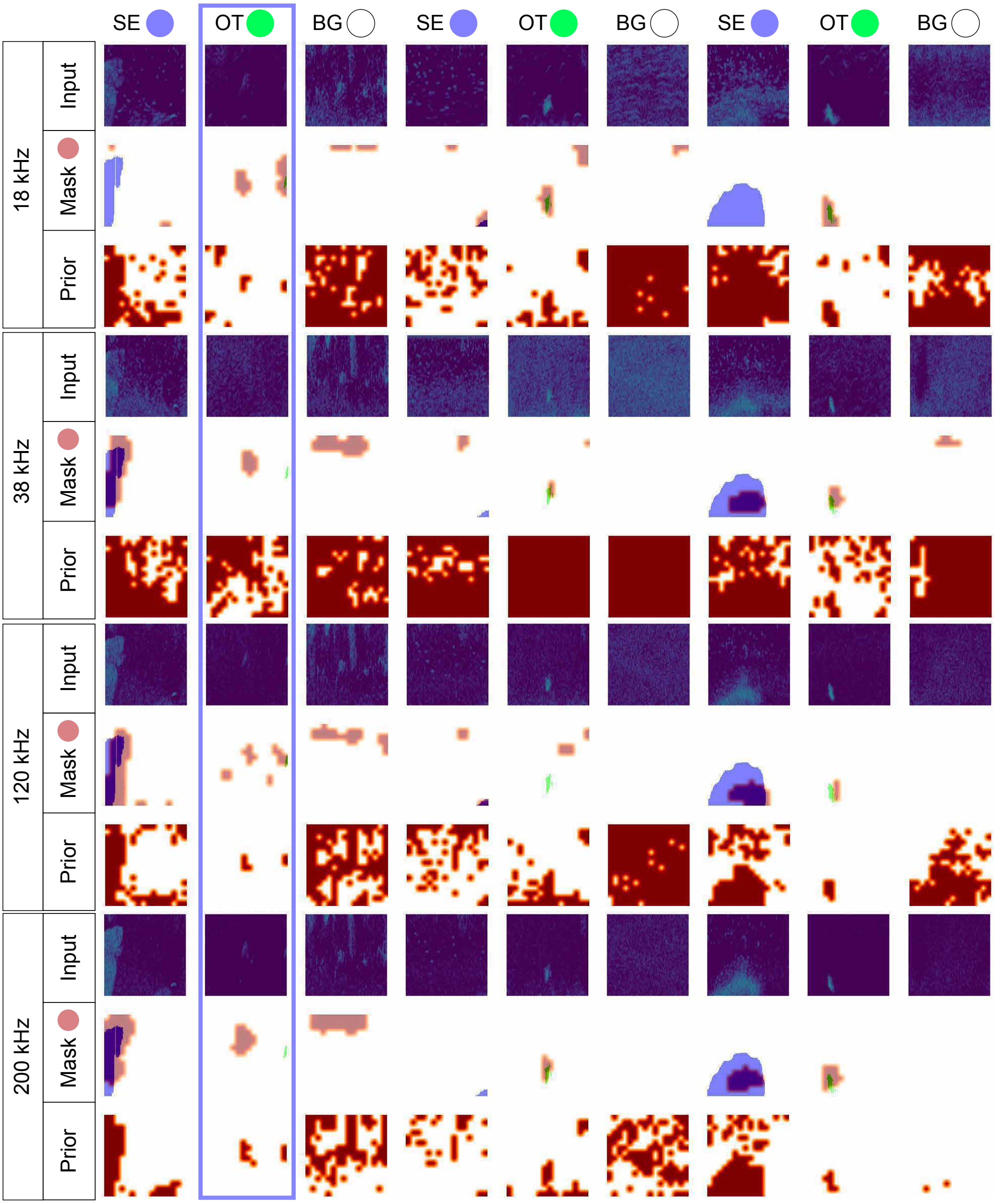


Figure B.13. Visualization of multi-Frequency echosounder Data, corresponding attribution masks M from DIB-X with $\beta = 0.005, \gamma = 0.005$, and the mask prior M_p per frequency channel.

Part IV

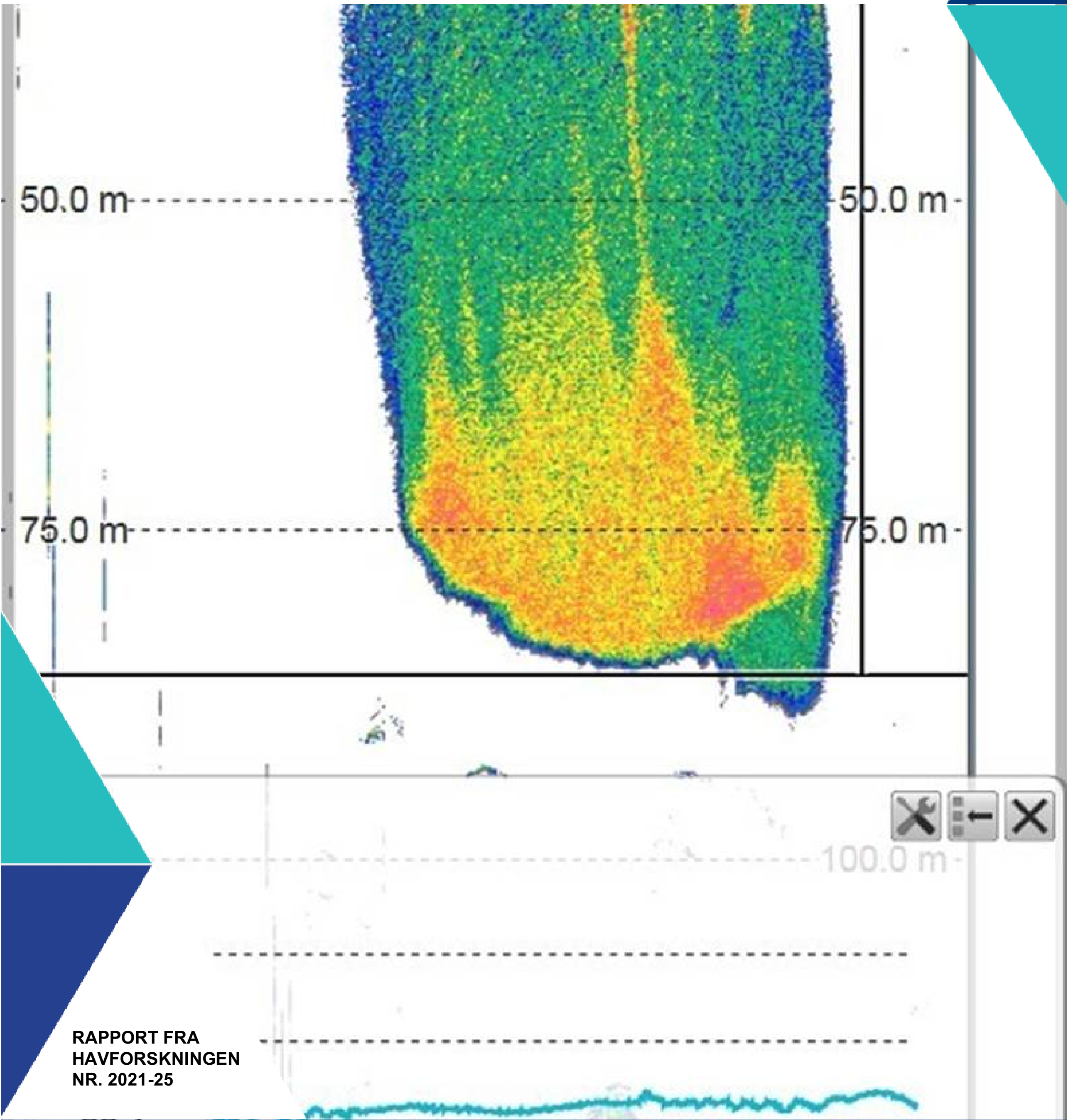
Other papers

12 | Paper 4

FISKERIAKUSTIKK OG AKUSTISK MÅLKLASSIFISERING

Rapport frå COGMAR/CRIMAC arbeidsmøte om maskinlæring og fiskeriakustikk

Nils Olav Handegard (HI), Lars Nonboe Andersen (Kongsberg Maritime), Olav Brautaset (Norsk Regnesentral), Changkyu Choi (UiT), Inge Kristian Eliassen (NORCE), Yngve Heggelund (NORCE), Arne Johan Hestnes (Kongsberg Maritime), Ketil Malde (HI), Håkon Osland (UiB), Alba Ordonez (Norsk Regnesentral), Ruben Patel (CODELAB), Geir Pedersen, Ibrahim Umar (HI), Tom Van Engeland (IMR) og Sindre Vatnehol (HI)



RAPPORT FRA
HAVFORSKNINGEN
NR. 2021-25

Tittel (norsk og engelsk):

Fiskeriakustikk og akustisk målklassifisering

Fisheries acoustics and Acoustic Target Classification

Undertittel (norsk og engelsk):

Rapport frå COGMAR/CRIMAC arbeidsmøte om maskinlæring og fiskeriakustikk

Report from the COGMAR/CRIMAC workshop on machine learning methods in fisheries acoustics

Rapportserie:	År - Nr.:	Dato:
Rapport fra havforskningen ISSN:1893-4536	2021-25	15.06.2021

Forfatter(e):

Nils Olav Handegard (HI), Lars Nonboe Andersen (Kongsberg Maritime), Olav Brautaset (Norsk Regnesentral), Changkyu Choi (UiT), Inge Kristian Eliassen (NORCE), Yngve Heggelund (NORCE), Arne Johan Hestnes (Kongsberg Maritime), Ketil Malde (HI), Håkon Osland (UiB), Alba Ordonez (Norsk Regnesentral), Ruben Patel (CODELAB), Geir Pedersen, Ibrahim Umar (HI), Tom Van Engeland (IMR) og Sindre Vatnehol (HI)

Godkjent av: Forskningsdirektør(er): Geir Huse Programleder(e): Frode Vikebø

Distribusjon:

Åpen

Prosjektnr:

COGMAR, CRIMAC

Program:

Marine prosesser og menneskelig påvirkning

Forskningsgruppe(r):

Bunnfisk, Pelagisk fisk, Økosystemakustikk

Antall sider:

25

Sammendrag (norsk):

This report documents a workshop organised by the COGMAR and CRIMAC projects. The objective of the workshop was twofold. The first objective was to give an overview of ongoing work using machine learning for Acoustic Target Classification (ATC). Machine learning methods, and in particular deep learning models, are currently being used across a range of different fields, including ATC. The objective was to give an overview of the status of the work. The second objective was to familiarise participants with machine learning background to fisheries acoustics and to discuss a way forward towards a standard framework for sharing data and code. This includes data standards, standard processing steps and algorithms for efficient access to data for machine learning frameworks. The results from the discussion contributes to the process in ICES for developing a community standard for fisheries acoustics data.

Sammendrag (engelsk):

Rapporten dokumentere eit arbeidsmøte I COGMAR og CRIMAC prosjekta om automatisk målklassifisering av akustiske data. Føremålet med arbeidsgruppa var todelt. I den første bolken gav partnarane ei oversikt over kva dei held på med innan fagfeltet. Først gav vi ei oversikt over bruk av maskinlæring på automatisk målklassifisering av akustikkdata. Maskinlæringsmetodar, og spesielt djuplæring, er i bruk på mange tilsvarande felt, i tillegg til identifisering av mål frå fiskeriakustikk. Den andre føremålet var å gje deltakarar med maskinlæringsbakgrunn ei innføring i fiskeriakustikk og diskutera korleis vi kan etablera data standardar for å kunne effektivt samarbeida. Dette inkluderer datastandardarar, standard prosesseringssteg og algoritmar for effektiv tilgang til data for maskinlæring Resultatet frå diskusjonane vart delt med arbeidet i ICES mot ein data standard innan fiskeriakustikkmiljøet.

Innhold

1 - Workshop format

The workshop was organized in two steps, were the first part was a mini conference on the use of machine learning methods on fisheries acoustics methods from the partners in the COGMAR and CRIMAC projects. The second part was a hands-on training session/hackaton on understanding, reading, and processing acoustic raw data.

The mini conference was a series of presentations of the approaches the different partners and institutions have used on fisheries acoustics data. The talks ranged from the recent advancement on using machine learning (ML) methods to the need for a framework for cooperation, data and algorithm sharing. The latter is linked to the ongoing efforts in ICES to develop a standard for acoustic data, and efficient connection to deep learning frameworks like Keras, Tensorflow and Pytorch, among others. The mini conference was held online November 1st, 2020.

The hackaton was organized December 7th- 11th 2020. The hackaton was a combination of small meetings, working in groups and training sessions on various aspects of reading, understanding and processing acoustic raw data.

2 - The mini conference

The mini conference was a series of presentations of the approaches the different partners and institutions have used on fisheries acoustics data. The talks ranged from the recent advancement on using ML methods to the need for a framework for cooperation, data and algorithm sharing. The latter is linked to the ongoing efforts in ICES to develop a standard for acoustic data, and efficient connection to deep learning frameworks like Keras, Tensorflow and pytorch, among others.

2.1 – LSSS and initial krill school detection by deep learning

Inge Eliassen and Junyong You gave a presentation of the Large Scale Survey System (LSSS) and presented initial work on using deep learning methods on krill school detections. They used screen shot images from LSSS as input to a U-net algorithm and Mask-RCNN and were experimenting with a RetinaNet architecture (Figure 1).

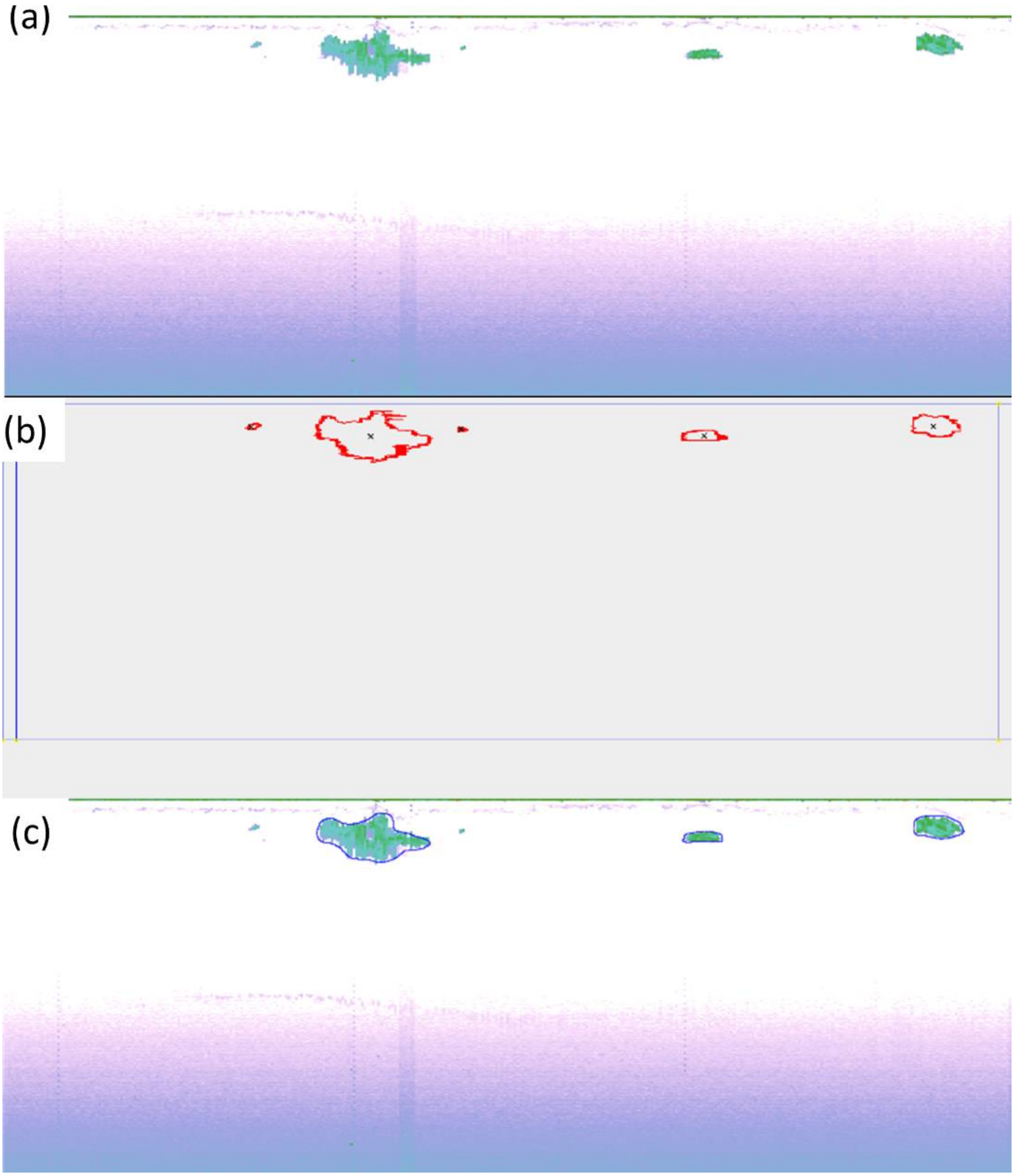


Figure 1. Krill school detection using a deep learning model. (a) Screenshot from LSSS, (b) manual annotation of krill schools, and (c) Mask-RCNN detection.

2.2 - Supervised learning and adding additional information to the classifier

Olav Brautaset presented the work on using the U-net algorithm on the sand eel data (Brautaset *et al.* , 2020). The continuation of this work includes addressing variations in ping rate, falsely detected high energy pixels, and how to include auxiliary information to the network (Figure 2). All these approaches show an improved performance of the network.

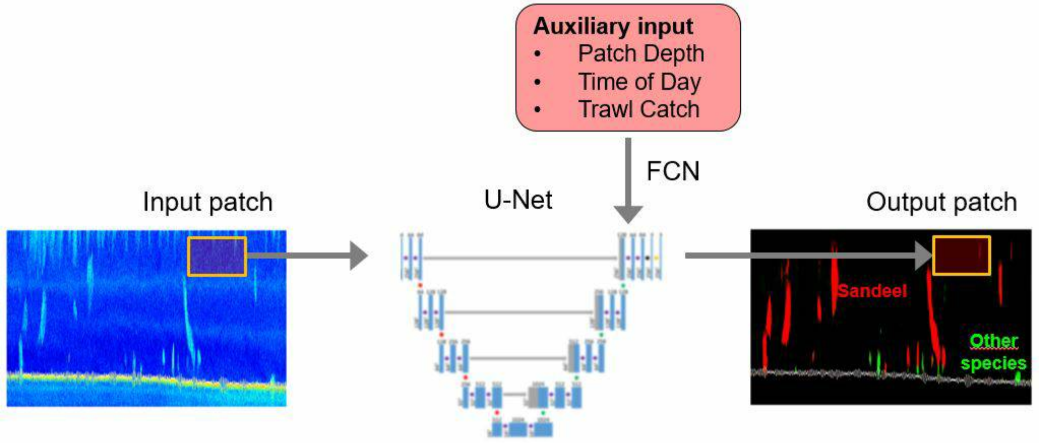


Figure 2. Including auxiliary information to the network for improved training and prediction.

2.3 - Semi-supervised deep learning approach using self-supervision

Changkyu Choi presented a semi supervised approach for classifying the sand eel data. He showed that a similar performance could be attained with only a subset of the labels. The idea is that all the data is used to learn a unsupervised representation for the data, and the labels are then combined in subsequent steps (Figure 3).

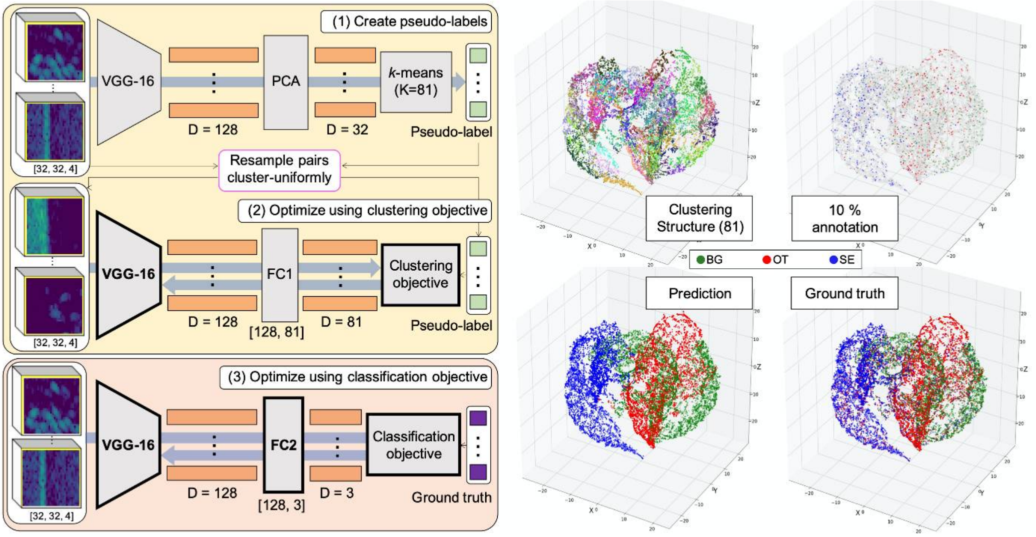


Figure 3. A semi supervised approach for classifying sand eel.

2.4 - Unsupervised deep learning

Håkon Osland is working on unsupervised methods to cluster acoustic data into different classes. this approach is valuable for learning the structure and representation of large amounts of data. He has been experimenting with generative adversarial networks for establishing a lower dimensional representation of the data.

2.5 - Clustering drop sonde data

Tom Van Engeland is working on fine detail data from a drop sonde system. He is working on clustering techniques to establish different acoustic classes from the data (Figure 4), and he is interested in whether the diversity of these classes can be used to address biodiversity in the area.

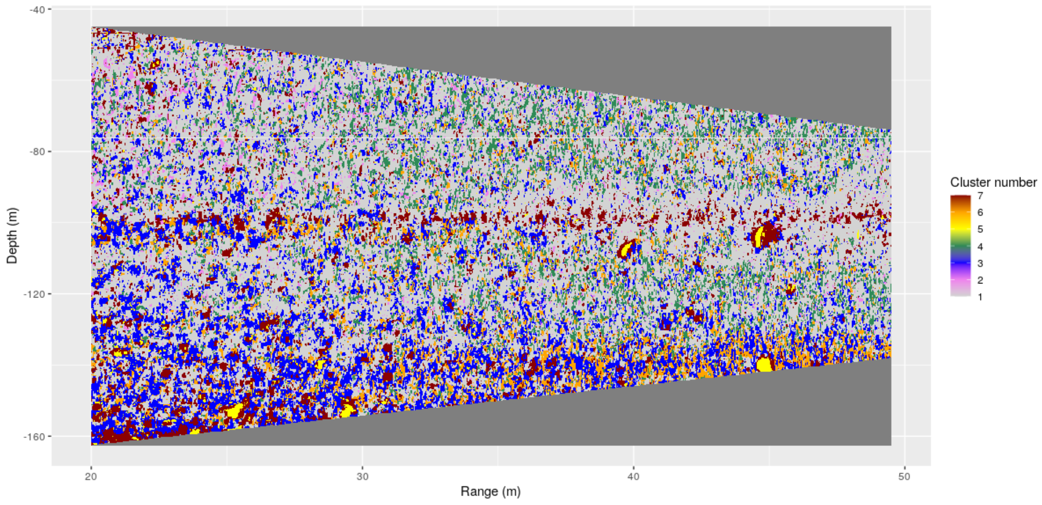


Figure 4. Different classes emerging when clustering the drop sonde data.

2.6 - IMR estimation workflow

Sindre Vatnehol and Ibrahim Umar presented the IMR data processing pipeline for fisheries advice. They emphasized the need to independent quality metrics on several levels, both on assessment results, survey indices, as well as finer scaled metrics. These metrics can be used for testing the sensitivity to different algorithms and parameters on and are useful for evaluating the different approaches in practice. They also emphasized the need for a data standard for the classification mask and the pre-processed data.

2.7 - Kongsberg processing pipeline

Arne Johan Hestnes presented the survey system that Kongsberg have been developing on top of their Kognifai platforms. We intent to use this platform for testing different classification algorithms. The platform supports docker images, and we intend to use that for testing and deploying different processing algorithms.

2.8 - General discussion

The work so far has been focused on traditional multiple frequency approaches using historical data. How to transfer this to broad banded data is an important step in the way forward, and the next steps will be to transfer that knowledge to cases where we start exploring the broad band spectrums.

The work so far has focused mostly on the Sand Eel survey since that have been made available for the participants, but it is important to move ahead with other surveys to ensure that we develop methods that are robust and scalable. The sand eel survey is also different to several other surveys in the sense that every school is manually labelled, whereas it is more common to allocate backscatter over a distance in other surveys. It is important that we include these other surveys to ensure that we develop methods that are general enough to tackle a broad range of problems.

The work presented have mainly focused on pixel based or patch-based predictions. This is the common approach for image analysis, but for acoustic trawl surveys, the backscatter itself is the key parameter. This means that low backscatter values are less important to get right than high backscatter regions. Weighing high intensity background images, like presented by Olav Brautaset, is an important step to reduce the overall error.

Different diagnostic tools need to be developed. These tools range from directly evaluating pixel wise annotations to the test set, via integrated backscatter over a distance, e.g. a transect, evaluating the performance metrics on the overall survey, and to effects on the assessment model results. These approaches can be used both for training, validation and testing. The latter is more relevant for testing since it requires larger computing resources. Sindre Vatnehol presented a few alternatives on how to move this forward, including different metrics to evaluate the consistency of a survey series.

The uncertainty in the acoustic target classification is not commonly included in survey estimates, but this constitutes an important source of uncertainty in the global estimate. The sensitivity to different classifications practices or algorithms can be analyzed and we can analyze the error propagation based on these uncertainties. This will be an important input to the survey estimation step, and the following assessment models.

2.9 - Discussion on collaboration and data standards

To make data available for the consortia and to ensure efficient collaborations across partners, the first step is to refine standards for exchanging data and algorithms between different computing modules. To steer the efforts, the existing pipeline will be used as a prototype. This pipeline consists of the manual classification from LSSS (Figure 1), the preprocessing and classification pipeline developed by IMR and NR (Brautaset *et al.* , 2020), the IMR data processing pipeline, and the Kongsberg processing and scheduling platform.

The data standard should contribute to the efforts of the ICES working group on Fisheries Acoustics, Science and Technology (WGFAST) in developing a common data convention[1], and details on how to contribute can be found there.

As a part of the ICES convention, a convention for interpretation masks is required. This information should contain the manual annotations and should cover content similar to the LSSS work files and the Echoview EV files. Code to read the LSSS work files exist[2]. A review of different data models has been performed[3], and a test implementation exist[4]. There is also code that run the predictions from the U-net algorithm, write the test version of the masks, and wrap it up in a docker image[5]. We need to test this and see if it is sufficient for our purposes. The goal is that everyone that creates models for acoustic target classification should write this format to allow for testing the predictions through the Kongsberg system and in the IMR processing pipeline.

The interpretation mask is used in combination with (preprocessed) raw data to generate the integrated backscatter. These data have an established data standard that we need to adhere to. Both Echoview and LSSS supports this standard, and it is the input to the IMR processing pipeline. We have code that can read the proposed interpretation mask convention and post it to LSSS, and then use the LSSS infrastructure to generate the output. The process is rather slow when applied to a full survey, and alternatives are to write the work files or to generate a standalone light weight integrator.

Python is one of the most commonly used programming languages within machine learning, and shared data access code base for accessing acoustic data and annotations are needed. The python libraries accessing the data formats should be cloud friendly, and the internal representation in python should be able to efficiently use common machine learning frameworks, like Keras, TensorFlow and PyTorch. Two python-based packages that can read acoustic raw data are available. Pyecholab[6] has support for low level reading of acoustic data, and echopype[7] that supports net cdf exports and zarr data files, that are a cloud friendly format. There is also a possibility to write python-based API's on top of the LSSS pre-processing code.

For implementation in Kongsberg and IMR's data processing pipeline, it is recommended to set up docker images for the different models and adhering to the input and output data models. A first version of a docker image for the preprocessor and the U-net classifier have been developed[8] That way, we can deploy the different models at a range of different platforms, both in the cloud and on platforms that are collecting data.

3 - The hackaton

3.1 – Objectives

To follow up on the recommendations from the one-day workshop, we set up a hackaton with the following objectives:

Objective 1: Learn how to read (and understand) acoustic data from single beam echosounders

The COGMAR and CRIMAC teams consists of people with skills across a wide range of fields, including fisheries acoustics, machine learning, statistics, etc, and the first objective was to get people familiar with the field of fisheries acoustics and to get hands on experience in reading and processing data. We use python as the language since most ML frameworks have good API's in python.

Objective 2. Code a pipeline from .raw to a gridded format

The first step in a processing pipeline (Figure 5) is to read the data and cast it into a format that can be read by modern machine learning libraries. An important part of this process is to provide input to the standardization process in ICES. The objective was to discuss how to best prepare the data for the ML framework. These objectives do not cover the full pipeline, and that will be the topic for future workshops.

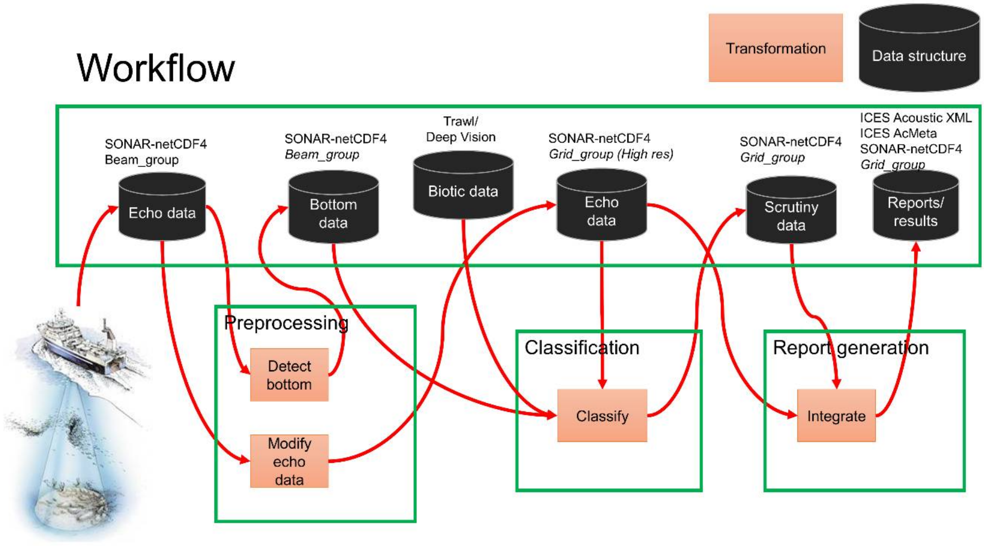


Figure 5. The suggested workflow. The black boxes indicates a data model and the orange rectangles denotes a processing step.

3.2 – Questions that you have had on acoustic data from echosounders, but never dared to ask

One of the main objectives was to bring participants with backgrounds in computer science up to speed on fisheries acoustics. To address this objective, we asked the participants after the symposium to list things that they did not understand or that were unclear. The following is a summary of answers to frequently asked questions that arose in this process.

3.2.1 - What is in the raw data?

The raw data from the split beam multi-frequency echo sounders (EK60) provide sampled backscattered energy together with athwartship (sideways) and alongship angles for each time step (ping). The angles are calculated by the phase difference between quadrants on the transducer face. The sampling rate and maximum range may vary between individual transducers, and may change between pings, and ping rate may vary over a survey transect. Data from individual pings may be missing due to intermittent system failures. As a result, the raw data will in general not fit directly to a time-range array. There are different ways of reading these data into python (Figure 6).

The EK80 split beam broadband echosounder has a continuous wave (CW, similar to EK60 but with higher sample rates) and a frequency modulated (FM) mode. The raw data from the EK80 in FM mode provides four channels of complex numbers, where each channel is from one quadrant of the transducer. The complex number denotes samples and the phase of the signal for each quadrant. In CW mode, these data can be processed to obtain data that correspond to the EK60 format, i.e. sv by range and angles from the phase differences between the channels. Note that the sample frequency is higher than for the EK60, but the data is still limited by the pulse length and remains similar. In FM mode, a chirp pulse is transmitted. This is a tone that change in frequency by time, and different sweeps can be configured. This also provides raw backscatter in four channels, but these data can be passed on to FFT algorithms to resolve the frequency domain. The data can be converted to a time-range-frequency grid per frequency channel after using the FFT.

The .raw files used by both EK60 and EK80 contains sequences of objects, referred to as ‘datagrams’. Each datagram starts with a length in bytes, the type (four ASCII letters, the last a digit signifying version), timestamp (two integers, alternatively, one long integer), the contents, and finally the length again as a sanity check. An EK60 file starts with a CON0 configuration datagram, followed by RAW0 datagrams each containing signal from one ping and one transducer/frequency. EK80 uses XML for configuration, and has an assorted number of new types (e.g. separate MRU datatype for heave/pitch/roll).

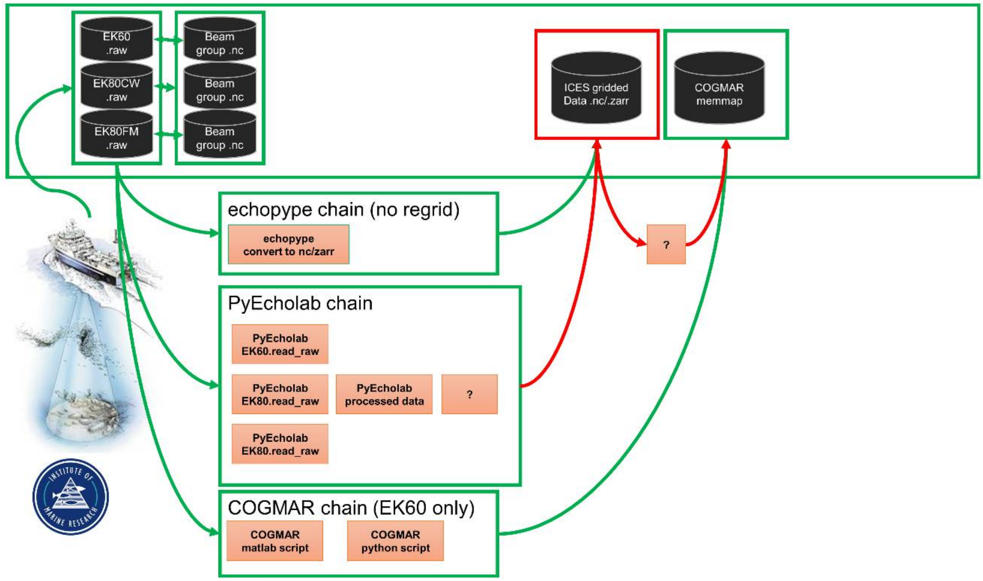
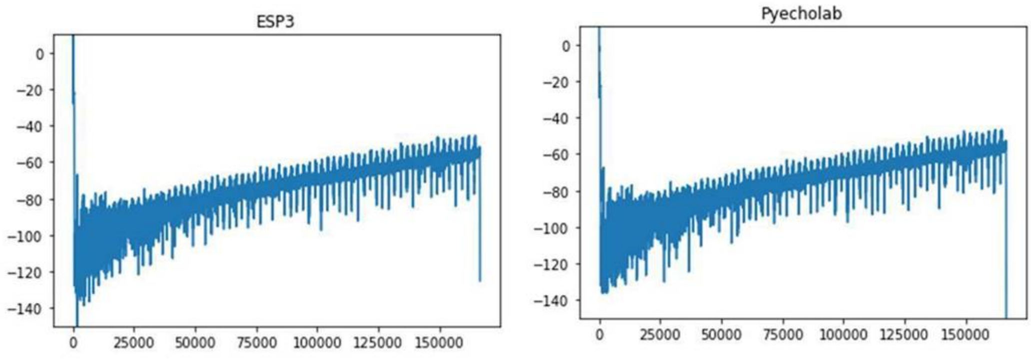


Figure 6. Different python/matlab packages tested for reading and regridding the data. Green lines are working software. The echopype package generates the zarr data but does not regrid data in cases where there are different pulse lengths between channels. The pyEcholab chain does not generate gridded data. The COGMAR chain works but relies on matlab functions that can only read EK60 data.

3.3 - Reading EK80

3.3.1 - Comparing EK80 readers between ESP3, pyecholab and LSSS

We tested different readers to see if there are differences in the initial data processing. An initial test showed some discrepancies (Figure 7).



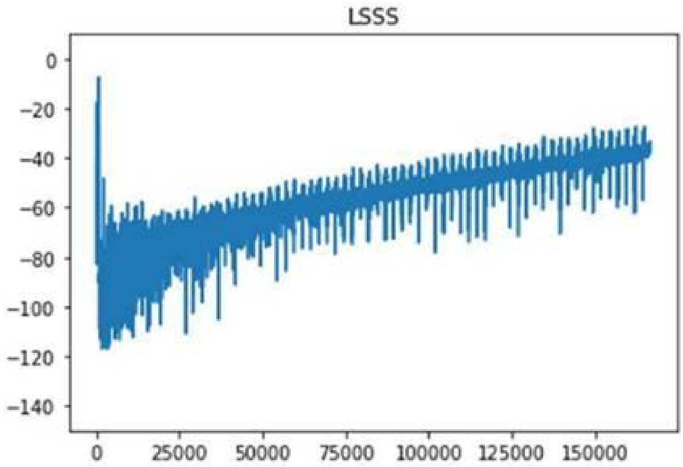


Figure 7. Comparison of Sv values from a single ping from ESP3, pyEcholab and LSSS.

The subgroup kicked off by installing and getting familiar with the pyEcholab package and LSSS. There were differences between the packages, and it seems like ESP3 and pyEcholab interpret the data incorrectly, and likely contain (some of) the same processing errors.

The background material for looking into this is the documentation of the EK80 interface[9].

An EK80 file with data collected in broadband mode (38, 70, 120, 200 kHz) and narrowband (18, 333 kHz) was selected as a demo data set for testing. This file was collected with GO Sars during the first CRIMAC cruise. The file was selected as it's reasonably small and collected using the latest version of the EK80 software (Nov 2020). Ping number 15 was selected for initial comparison of Sv from LSSS and pyEcholab. The first 1000 samples show some discrepancies between LSSS and pyEcholab (Figure 8).

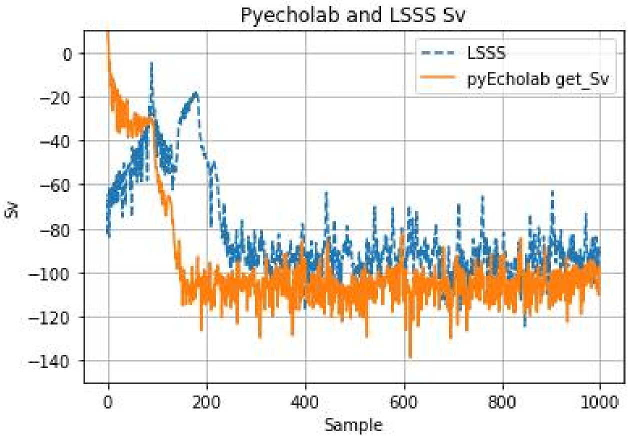


Figure 8. Comparison of the first 1000 samples of the demo file as read by LSSS and pyEcholab. (get_samples in Python) and pyEcholab (get_Sv).

The complex values from the EK80 files were read per sector by LSSS and pyEcholab, allowing for a step by step process to identify discrepancies. The Raw complex values (Figure 9), as read by pyecholab corresponds to LSSS and seems ok. A systematic comparison of the LSSS implementation and the pyEcholab code (EK80.py) was then performed, and in the following analysis three discrepancies were found in calculations of pulse compressed Sv compared to Simrads description on how to calculate pulse compressed Sv:

1. pyEcholab uses gain at the nominal frequency. This should be at the centre frequency.
2. pyEcholab uses psi at the nominal frequency. This should be at the centre frequency.
3. \tau_eff uses the signal, and not the autocorrelated signal.

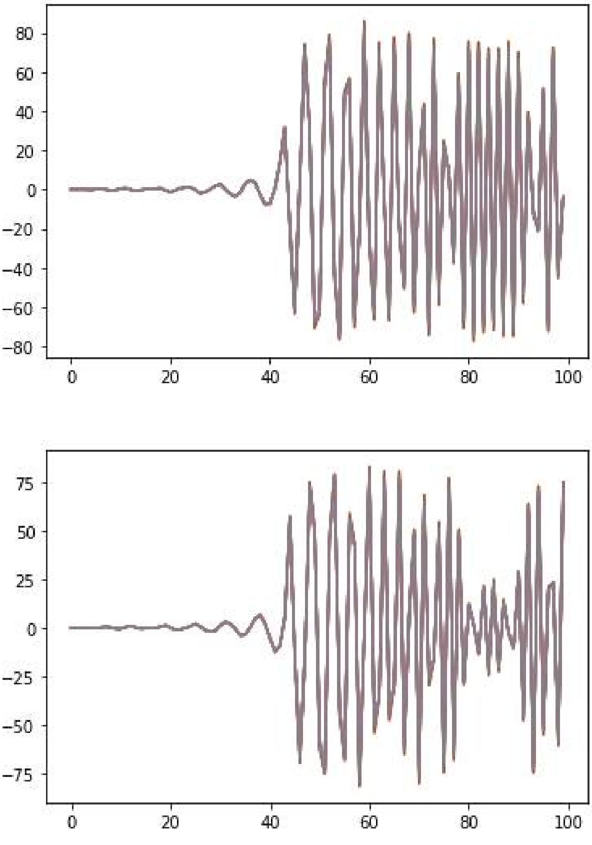


Figure 9. The first 100 samples from the real (left panel) and complex (right panel) values from quadrant 1.

By applying these three changes to the pyEcholab code base, the results from LSSS and pyecholab are similar (Figure 10). The modification was implemented in an “ad hoc” manner for the Hackaton, and needs a proper implementation, reading the relevant values from the raw file. The most recent update of the pyEcholab package have adopted these changes.

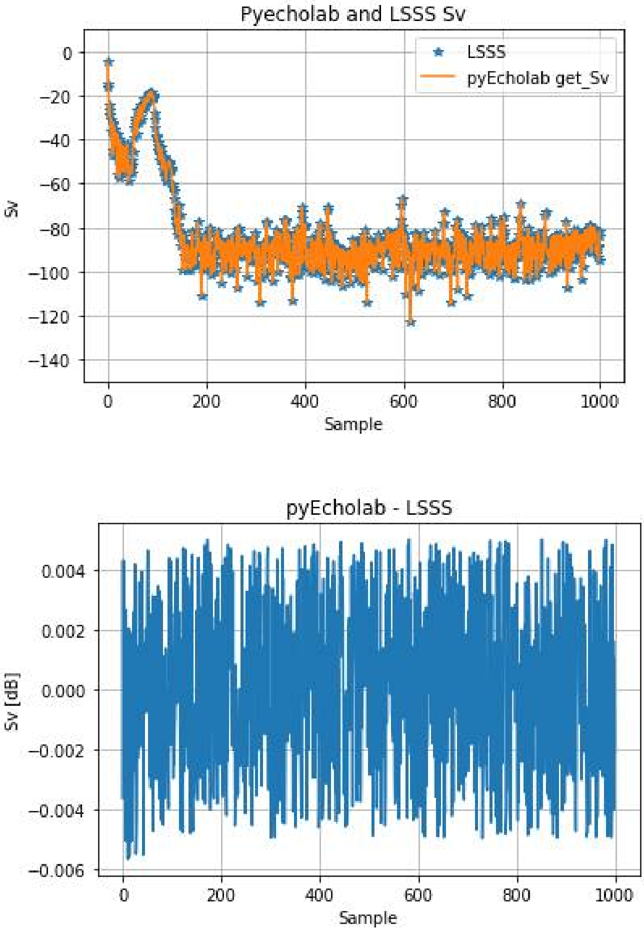


Figure 10. (left panel) Side-by-side comparison between and (right panel) difference between LSSS and pyEcholab after modifying pyecholab.

In addition there are two differences between LSSS and pyEcholab, which are not errors but more conventions to be agreed upon:

1. Should negative range values be plotted? LSSS keeps all data while pyEcholab only keeps positive values
2. pyEcholab : range=max(1,range), LSSS: range=max(sampleDistance, range)

3.4 - A ML friendly convention for echosounder data

Multifrequency echosounder data are stored in arrays per frequency, but machine learning libraries typically work on tensors. There has been an effort within ICES to move forward with a standardized format, and this part of the workshop reviewed the proposed standard for raw and processed echsounder data and provided input to the ICES process.

3.4.1 - The COGMAR “Echogram” class

The COGMAR project used the Sand eel survey as a test case and developed a ML friendly proprietary data format for the raw files and the label files from LSSS. Each pair of .raw and associated label is in this context called an “echogram”. This is a working processing pipeline and serve as a starting point for the discussion. The intention is to adapt this pipeline to the new format. The converted data from each raw file (echogram) are stored in separate directories (Table 1).

The echogram class regrid the data into a tensor based on the grid for the 38kHz channel. Although the different frequencies are stored as separate files, they are aligned in ping and time.

Table 1. The individual files from one pair of work and raw files.

	File name	File type	Data structure	Data type	Description
Acoustic data	data_for_freq_18	.dat	array(y, x)	float	Echogram data interpolated onto a (range, time) grid, common for all frequencies. Not heave corrected. Stored as numpy.memmap.
	data_for_freq_38	.dat	array(y, x)	float	
	data_for_freq_70	.dat	array(y, x)	float	
	data_for_freq_120	.dat	array(y, x)	float	
	data_for_freq_200	.dat	array(y, x)	float	
	data_for_freq_333	.dat	array(y, x)	float	
Labels	labels	.dat	array(y, x)	int	Species index mask. Heave corrected. Stored as numpy memmap.
	labels_heave	.dat	array(y, x)	int	Species index mask. Not heave corrected. Stored as numpy.memmap.
Utility data	objects	.pkl	list(dict)	*	*See description.
	range_vector	.pkl	array(y)	float	Vertical distance to ship.
	time_vector	.pkl	array(x)	float	Time stamp for each ping.
	heave	.pkl	array(x)	float	Relative ship altitude above mean sea level.
	depths	.pkl	array(x, f)	float	Vertical distance to seabed for each frequency. Seems to be heave corrected.
	seabed	.npy	array(x)	int	Vertical distance to seabed (in-house estimate from acoustic data).
Metadata	shape	.pkl	tuple(2)	int	Shape of range-time grid.
	frequencies	.pkl	array(f)	int	Available frequencies.
	data_dtype	.pkl	-	str	Data type of the acoustic data.
	label_dtype	.pkl	-	str	Data type of the label masks.

*objects.pkl contains a list of schools. Each school is a dictionary where:

- indexes: (list) Echogram indices for the school
- bounding_box: (list) Echogram indices for bounding box corner coordinates for the school
- fish_type_index: (int) Species index for the school
- n_pixels: (int) Echogram pixel count for the school
- labeled_as_segmentation: (bool) False if ‘indexes’ is a rectangle, True else

We have defined a python class “Echogram” that can be called to create an echogram object for any echograms. Calling the Echogram class, an echogram object is initiated with convenient attributes (e.g. the echogram’s name, shape, range_vector, time_vector, schools, heave, etc.).

Each echogram object has methods for reading the acoustic data and labels. These data are read from memory map files (numpy.memmap), which enables reading patches of the data without loading the entire file into memory. This allows for efficient data loading, e.g. for training neural nets on small echogram patches. Each echogram object is also equipped with a method for plotting the acoustic data, labels, classifier predictions, etc.

The class and data files are highly efficient when training CNNs, but splits the data into “echograms” that has no physical meaning other than the file size set for storing the data. The question is if we can define a format that are equally efficient, store one survey as one continuous “echogram”, and follow the ICES standard.

3.4.2 - Preprocessing

The raw data is organized as a data from individual pings, but the data can be cast into a regular time-range-frequency grid without any regridding if the ping and pulse lengths are similar.

In some cases, the data sets have a different resolution in time and range. This prevents us from casting the data into a time-range-frequency array. Discrepancies in time may be caused by ping dropouts from some transducers, or, if sequential pinging between echosounders have been used, different time vectors. Different range resolutions occur when the pulse lengths are different (EK60) or different averaging intervals are used (EK80 CW). In these cases, the data needs to be regridded to fit a tensor or the echogram class described above.

There are indications that regridding the data may affect performance, and ideally it should be avoided to the extent possible.

For regridding in time, our test algorithm aligned the ping in time using match_ping method of the ping_data in pyecholab. We insert NaN’s where there are missing pings in one or more frequencies. The test implementation fit the data onto the ping vector for the main frequency, but a better approach may be to use the union among all the pings as the target grid for the time resolution.

For regridding in range, a first order conservative regridding should be used. This preserve the echo energy when integrating across range. We tested the following approach. For each ping there is a source range vector ‘r_s’, the source sv_s vector, a target range ‘r_t’ coming from the “master” frequency, and the resulting target sv_t (Figure 11). The source range vector can either be of higher or lower resolution than the target range vector. The target range vector is typically the main frequency used for the echo integration, but it can be any of the frequencies. The mapping is a linear mapping between the source and target and the weight matrix is sparse with most weights close to the diagonal. An example can be found here https://github.com/CRIMAC-WP4-Machine-learning/CRIMAC-preprocessing/blob/NOH_pyech/regrid.py , but any mapping that conserve the mass between the grids could be used and standard packages may be used for this purpose.

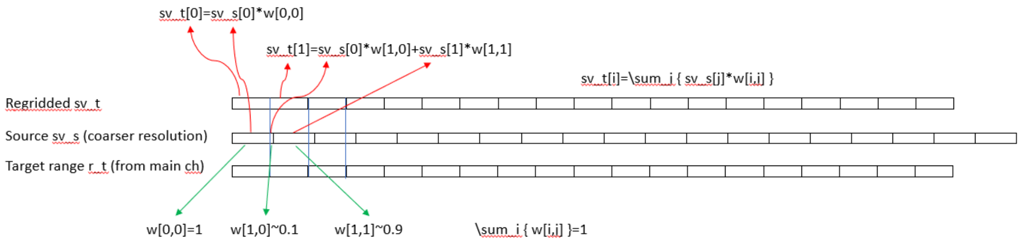

sv_t[0]=sv_s[0]*w[0,0]
sv_t[1]=sv_s[0]*w[1,0]+sv_s[1]*w[1,1]
sv_t[i]=\sum_i { sv_s[j]*w[i,j] }
Regridded sv_t
Source sv_s (coarser resolution)
Target range r_t (from main ch)
w[0,0]=1
w[1,0]~0.1
w[1,1]~0.9
\sum_i { w[i,j] }=1

Figure 11. Converting the data from a coarser source resolution to a higher target resolution. The algorithm will work both ways.

There are several approaches to grid the data, and each step require more pre-processing with potential data loss. There was a thorough discussion on how much processing could be allowed before passing it onto a ML framework. The idea is that most of these steps can be potentially better handled by the ML framework than ad hoc decision on the pre-processing. The conclusion was to define a set of pre-processing steps, where the first step would be lossless, followed by steps that are near-lossless, e.g. only resample in rare occasions, to more fully gridded version where both time/distance travelled and depth/range a gridded to a uniform grid. If a gridding operation is performed, it should also be labelled in the data. For the latter the resolution may be set similar to the reports that goes into the ICES acoustic data vase, efficiently supporting the whole processing pipeline.

The outcome of this discussion is documented here:

https://github.com/ices-publications/SONAR-netCDF4/issues/33

And in the following pull request:

https://github.com/ices-publications/SONAR-netCDF4/pull/34

3.4.3 - The ICES sonar-netcdf convention

After casting the data into an array/tensor, the convention needs to support storage. To learn more about the different options, we set up some test code for writing NC fields. The echopype package can write both nc and zarr files, but the convention needs to catch up. The objective of this sub task was to go through the steps and write up an np array to .nc to develop the gridded group in the standard.

The test method ek2nc() writes a 3d numpy array to a netcdf4 file. The method takes three parameters:

- variable : the actual 3d array with dimensions in order (time, range/bins, channels)
- dims: a list of dimension variables in the same order as indicated for the “variable” parameter above
- file: the filename as a text string

Gradually the method will be extended to comply with the Sonar-NetCDF standard. Since the parameters to the implemented method do not contain metadata information, this must be supplied manually. After a preliminary comparison between the information content of the EK80 object in Python and the Sonar-NetCDF convention, it seems that part of the data will always have to be provided manually, unless the PyEchoLab package can be updated to draw the information from the raw files. This would be part of a strategy where the ek2nc method receives an EK80 object, possible regridding is done inside the method, and all the meta-information is transferred from the EK80 object to the NetCDF file.

What order should the axis be in? Local in time-space first? Time last since that may cover a full survey? See discussion in this thread: https://github.com/ices-publications/SONAR-netCDF4/issues/33. In theory the ordering of the dimensions, time, range/bins, and beams, can have an impact on I/O performance because a 3D array is essentially a (preferentially contiguous) 1D (chunked) array in memory and on disk. As a result, performance of accessing subsets along one dimension can vary with dimension, because the file pointer may have to jump back and forth if reading along a suboptimal dimension.

Numpy use by default row major ordering of array cells in memory, like C; i.e. C-major ordering), but can also store in column major ordering (like Fortran, Matlab and R; i.e. F-major ordering). NetCDF also uses row-major ordering, implying that there is a 1-1 association in the cell ordering between Python and NetCDF.

for i in ...: for j in ...: for k in ...: A[i][j][k] = ... # efficient in Python, inefficient in Fortran

In NetCDF's row-major ordering the first dimension a of an array A with dimensions (a,b,c) is the slowest and c the fastest changing dimension. This means that, if we want to load entire water column profiles of backscatter but only over limited periods of time, time should be the first and slowest dimension (in NetCDF terms also called the record dimension). In classical NetCDF (< 3.6) it was mandatory to put this dimension FIRST.

As to which dimension to choose as second, several considerations need to be taken into account.

- We may cut off part of the profiles that are under a bottom (for instance if a technician is not paying attention during a survey to adjust the transducer ranges over the continental slope). But to do this we have to look at the data first, which implies that we have to load it (or at least one channel). Subsetting along the beam range is in theory quicker if the beam dimension is chosen as the second dimension.
- We may decide to use only a subset of channels, in which case we can optimize by putting the channel dimension second.

NetCDF uses chunking. Chunks are smaller units of data that can be at random position in a file and that are optimally accessed. Chunking and chunk caching (~ keeping in memory what is regularly used) are features that are by default taken care of by the NetCDF software layer. Chunk size and layout are determined based on the sizes of the fixed dimensions. This makes a decision on the order of the fixed dimensions less urgent. Optimizing chunking for one type of data access is likely to worsen other types of data access. It may also be worthwhile to consider how the data is used in Python.

In this early stage, the implemented function (github) to write an Numpy array to NetCDF requires the unlimited time dimension to be the first, followed by the range/bin dimension, and the channel dimension.

This needs to be aligned with the data standard.

3.4.4 - Fitting EK80 FM data into the gridded structure

The EK80 raw format stores the raw sample data and allows for more processing. The volume backscattering (or Target Strength) compressed over the operational frequency band at each sample can directly follow the proposed structure of EK60 and EK80 CW.

Volume backscattering (or Target Strength) as a function of frequency (frequency index m) requires additional processing, which is not implemented in echoPype. There are also choices made in the processing that have implications (e.g. length/distance of the FFT window). There are a few possibilities to do this.

- Converting complex data to sv at centre frequency. This is the simplest option, but it does not take advantage of the extra information in the FM data.
- Converting complex data to sv at multiple frequency bands. This will extract some of the information in the FM data. The result will look as multiple CW frequencies.
- Converting complex data to continuous sv(f) (or TS(f)) for ranges of samples. But how should the sample ranges be chosen? This could be done for the samples of a single target (fish), or for the depth range of a school.
- Using the complex sample data directly in the machine learning step. This could also need all the meta-data used for converting to sv, such as pulse duration, transmit power, calibration data, etc.

This question has been raised for ICES, and the discussion can be found here: https://github.com/ices-publications/SONAR-netCDF4/issues/33#issuecomment-745347642

3.4.5 - Converting NC data to memmaps

The echogram class used in the COGMAR project currently relies on the data being stored as numpy memory maps (see section above). This file format allows the user to read only a small part of a large file, without loading the entire array into memory. The netCDF files will need to be interfaced with the echogram class for the COGMAR pipeline to work.

One option is to convert the data from the netCDF to the COGMAR files. A conversion script can easily be created, using the netCDF4 python package, to read the relevant parts of the netCDF files. This will be a simple first solution to extend the pipeline to other surveys being prepared in the netCDF format.

Another option is to rewrite the echogram class and read the netCDF files directly during training. The echogram class is used to repeatedly fetch small subsets (contiguous ping-range patches) of the echogram for the training of a neural network. The netCDF allows subsetting, but the access speed must be evaluated. To avoid slowing down the training, this task needs to run quickly and be memory efficient. Instead of using the netCDF4 python package, the xarray python package seems to be a better option. This option needs to be tested.

Being integrated with Dask, xarray allows computations on a dataset similarly to a numpy memory map, i.e. there is no need to load the entire array into memory in order to access a subset. The array is divided into smaller chunks, where the shape of the chunks is determined by the user. Selected chunks may be loaded into memory in an explicit conversion step. The chunk shape may be selected so that it matches the patch size currently used in the training of the neural network. The patch selection could be rewritten so that only one chunk is loaded when selecting a patch. Alternatively, the patch selection may be kept as is, as loading multiple chunks into memory should not affect the performance much.

An example of Xarray-way of chunked read a NetCDF4 file can be found here: https://github.com/iambaim/crimac-hackathon/blob/main/nctomap.ipynb (notebook) or https://github.com/iambaim/crimac-hackathon/blob/main/nctomap.py (Python code).

4 - Concluding remarks

There are several tasks that needs attention on a shorter and longer time scale:

- A data convention for preprocessing and gridding the acoustic data is needed
- Regridding should be avoided if possible, but is needed in cases where pulse lengths are set different (CW)
- The convention for preprocessing FM data needs to be developed
- The preprocessed data should be directly usable by the deep learning frameworks
- There is a need to address data provenance throughout the processing pipelines
- There is a need to develop a standard for annotations
- Existing ML models should be interfaced to the the standard
- Build further processing algorithms should be built on to of the standard

Both the COGMAR project and CRIMAC will follow up on these recommendations, and contribute to the international processes aiming at developing a common community standard for these instruments: https://github.com/ices-publications/SONAR-netCDF4

5 - References

Brautaset, O., Waldeland, A. U., Johnsen, E., Malde, K., Eikvil, L., Salberg, A.-B., and Handegard, N. O. 2020. Acoustic classification in multifrequency echosounder data using deep convolutional neural networks. ICES Journal of Marine Science, 77: 1391–1400.

Korneliussen, R. (Ed). 2018. Acoustic target classification. ICES Cooperative Research Report, No. 344: 104pp.

[1] https://github.com/ices-publications/SONAR-netCDF4
[2] https://github.com/nilsolav/LSSSreader
[3] https://docs.google.com/document/d/1F5ub9-ElnGWgoFzOhwrNiAB6fZRhKI8Nw6FskMhzl0g/edit#heading=h.ihw2gdxqw9td
[4] https://github.com/nilsolav/EchosounderNetCDF
[5] https://github.com/CRIMAC-WP4-Machine-learning/CRIMAC-classifiers-unet
[6] https://github.com/CI-CMG/pyEcholab
[7] https://github.com/OSOceanAcoustics/echopype
[8] https://github.com/CRIMAC-WP4-Machine-learning
[9] https://www.simrad.online/ek80/interface/ek80_interface_en_a4.pdf

HAVFORSKNINGSINSTITUTTET

Postboks 1870 Nordnes
5817 Bergen
E-post: post@hi.no
www.hi.no

13 | Paper 5

Short-Term Load Forecasting with Missing Data using Dilated Recurrent Attention Networks

Changkyu Choi[1], Filippo Maria Bianchi[2], Michael Kampffmeyer[1], and Robert Jenssen[1]

[1]UiT The Arctic University of Norway
[2]NORCE Norwegian Research Centre

Abstract

Forecasting the dynamics of time-varying systems is essential to maintaining the sustainability of the systems. Recent studies have discovered that Recurrent Neural Networks (RNN) applied in the forecasting tasks outperform conventional models. However, due to the structural limitation of vanilla RNN which holds unit-length internal connections, learning the representation of time series with *missing data* can be severely biased.

We propose *Dilated Recurrent Attention Networks* (DRAN), a robust architecture against the bias from missing data. This has a stacked structure of multiple RNNs, with each layer leveraging a different length of internal connections to incorporate previous information at different time scales, and updates its output state by a weighted average of the states in the layers. In order to focus more on specific layers that carries reliable information against missing data bias, our model leverages attention mechanism which learns the distribution of attention weights among the layers. The proposed model achieves a higher forecast accuracy than conventional ones from two benchmark time series with missing data that include a real-world electricity load dataset.

1 Introduction

An inaccurate forecast may pay an expensive price for financial and social deterioration which are unanticipated [3, 4]. Since the reliability of the forecast has a strong impact on the economic feasibility of industry [1], Short-Term Load Forecasting (STLF) in time-varying systems has been explored actively. Still, this is a difficult task as it depends on not only the nature of the system but also external influences. In the case of electricity consumption, we initially take distinct time dependencies into account as a nature of the system, namely intra-day, intra-week, and across different seasons [8]. Some external influences, such as calendar effects and rapid change of meteorological conditions, add irregularities on top of it [10].

Complex load patterns driven by the in- and external influences restrict the forecast to a given degree with conventional approaches, as they require strong statistical assumptions. RNN, a member of neural networks known for more flexibility with little prior assumptions, has become a standard framework for STLF tasks after outperforming conventional forecasting models that include AutoRegressive Integrated Moving Average (ARIMA) [4].

Missing data is a classical but critical problem in data analysis. They arise due to imperfect data collection, or various types of censoring [13]. Their possible effect on the results is seldom quantified despite the fact that they are a likely source of bias [15]. RNN can contribute to mitigating the bias from missing data by relying more on the previous information rather than the current missing data, as the internal connections play a role of memory. In addition, this learns rich information from the missing pattern, re-

https://doi.org/10.7557/18.5136

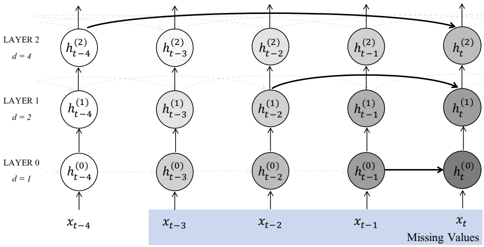


Figure 1: Unfolded graph of the dilated RNN with layer $L = 3$, DRNN(3). Consecutive four values $\{x_{t-3}, x_{t-2}, x_{t-1}, x_t\}$ in the blue window are missing. The gray-scale color of the RNN unit represents the degree of the bias from the missing values.

ferring to *informative missingness* [6]. Several RNN studies successfully attain the classification task with missing data [6, 12], however, there is a room for the study of STLF tasks that focuses on missing data.

We propose DRAN, a novel framework tailored for STLF tasks with missing data. This inherits the properties of Dilated RNN (DRNN) [5], featured by a multi-layer and cell-independent architecture, where each layer has a different internal connection, referred to *dilation*. To the best of our knowledge, this is the first STLF paper that applies RNN on the missing data problem. The model we suggest is readily applicable to other types of tasks but we limit ourselves to STLF tasks in this paper.

2 Dilated Recurrent Neural Networks

DRNN [5] is featured by *dilation* $d^{(l)}$, which is defined by initial length d_0, and base M. It is specified in Equation (1), where layer $l = 0, 1, \cdots, L-1$, state $\mathbf{h}_t^{(l)}$, and input x_t corresponding to layer $l = -1$.

$$\begin{aligned} \mathbf{h}_t^{(l)} &= f(\mathbf{h}_{t-d^{(l)}}^{(l)}, \mathbf{h}_t^{(l-1)}) \\ d^{(l)} &= d_0 M^l \end{aligned} \tag{1}$$

This enables the capture of multiple time dependencies and aggregate multi-scale temporal context into output. This provides more flexibility and capability in learning representation of the time series. The literature suggests to let $d^{(l)}$ have exponentially increasing length, as introduced in WaveNet [14] and Dilated CNN [16].

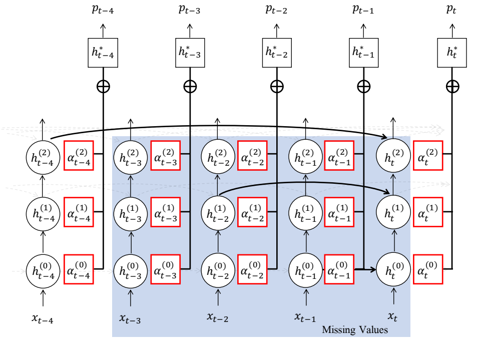


Figure 2: DRAN with layer $L = 3$, DRAN(3), with dilation $d^{(0,1,2)} = \{1, 2, 4\}$.

Role of Dilation towards Missing Data

Figure 1 represents how dilations operate in a missing window that consists of consecutive missing values $\{x_{t-3}, x_{t-2}, x_{t-1}, x_t\}$ represented by a blue box in the figure. As input values within the missing window are biased, it is reasonable to argue that a less number of the state update will protect the networks from the bias. Dilation is closely linked with the update frequency of the state $\mathbf{h}_t^{(l)}$. By comparing two dilations in LAYER 0 and LAYER 2 in Figure 1, it is evident that exploiting layers with longer dilation more in the missing window will reduce the update frequency of the state.

3 Dilated Recurrent Attention Networks

Figure 2 illustrates DRAN with layer $L = 3$ that improves DRNN(3) in Figure 1. The idea of DRAN is to leverage the attention mechanism [2] in regulating the exploitation of the layers when dealing with missing data. The attention mechanism is to make specific internal states contribute more to the output state, where a weighted average is the general form

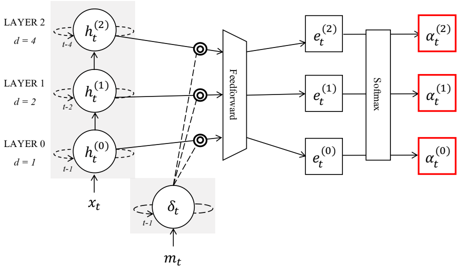


Figure 3: Schema of constructing attention for DRAN(3). The attention $\alpha_t^{(l)}$ is obtained by the score $e_t^{(l)}$ applied by a softmax function. The score is derived by the concatenation of missing history δ_t and the state $h_t^{(l)}$ processed by feedforward neural networks.

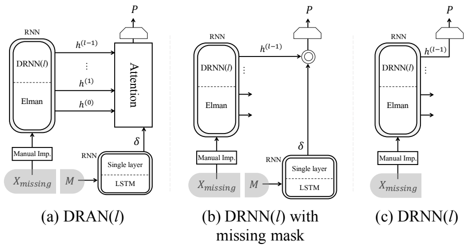
of the contribution. We define the trainable weights $\{\alpha_t^{(l)}\}$ as attention parameters.

We argue that DRAN simultaneously learns the representation of the states $\{\mathbf{h}_t^{(l)}\}$ and the distribution of the attention weights $\{\alpha_t^{(l)}\}$ over the layers in order to determine the exploitation of the layers with different dilations.

Depicted in Figure 3 and Equation (2), the construction of attention parameters that DRAN utilizes is unique and is inspired by two different methods, the attention mechanism [2] and missing history setting from GRU-D [6].

$$\begin{aligned} \alpha_t^{(l)} &= \frac{exp(e_t^{(l)})}{\sum_{k=0}^{L-1} exp(e_t^{(k)})} && :\ softmax \\ e_t^{(l)} &= g(\boldsymbol{h}_t^{(l)}; \boldsymbol{\delta}_t) && g\!:\ FFNN \end{aligned} \tag{2}$$

The attention parameters $\{\alpha_t^{(l)}\}$ are derived from the scores $\{e_t^{(l)}\}$, processed by the softmax function so that they have values within the interval $[0,1]$ and the sum over the layers is one. The scores $e_t^{(l)}$ play a role in incorporating current $\boldsymbol{h}_t^{(l)}$ and $\boldsymbol{\delta}_t$, representing the state at each layer and the missing history of input x_t respectively. Scores are derived by the concatenation of these two vectors, processed by a feedforward neural networks(FFNN).



(a) DRAN(l)

(b) DRNN(l) with missing mask

(c) DRNN(l)

Figure 4: Model comparison: (a) DRAN(l); (b) DRNN(l) with missing history binary mask; (c) DRNN(l). Elman RNN refers to the vanilla RNN. Every model has input with missing values $\mathbf{X}_{missing}$. The effect of attention is compared by the model (a) and (b), where model (b) concatenates the output states of two RNNs. Model (c) are suggested to see the effect of missing mask by comparing with model (b). **M** *and* **P** *represent binary mask and forecast respectively.*

$$\begin{aligned} \boldsymbol{\delta}_t &= f(\boldsymbol{\delta}_{t-1}, m_t) \qquad f\!:\ external\ RNN \\ m_t &= \begin{cases} 1, & \text{if } x_t \text{ is observed} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

Missing history $\boldsymbol{\delta}_t$ is the state of an external/small RNN. It is derived from binary mask time series m_t in Equation (3), processed by other RNN which are trained jointly, such as LSTM.

4 Experiments

The experiments are designed to compare DRAN(l) in Figure 4(a) with two reduced models, reduction of the attention unit in Figure 4(b), referring to DRNN(l) with *missing mask*, and reduction of the external RNN(LSTM) in Figure 4(c), referring to DRNN(l). Two baseline models, Gated Recurrent Unit(GRU) [7] and ARIMA(p,d,q), are chosen and compared with the three models mentioned above. The order of ARIMA(p,d,q) is carefully selected by following commonly used practices for the design of the coefficients[1].

[1] `https://people.duke.edu/~rnau/arimrule.htm`

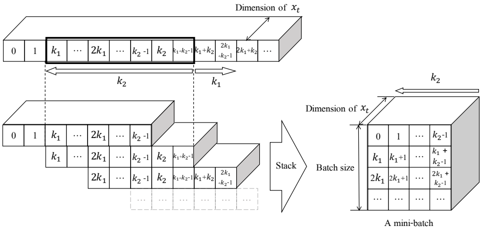
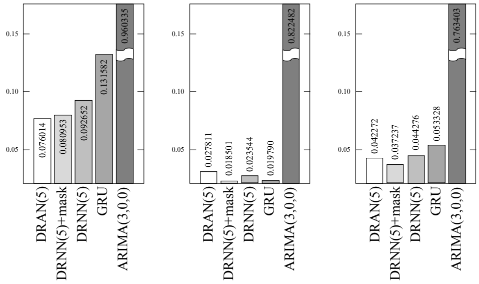


Figure 5: Formulation of a mini-batch for tBPTT(k_2, k_1).

We analyze both a synthetically generated time series; Mackey-Glass (MG) system, and a time series from real-world load data from a public dataset; GEFCom 2012 competition [11], in order to provide controlled and easily replicable results for the architectures under analysis. MG dataset is given without missing values, hence, we assign missing values in the time series. To observe the performance when values are missing consecutively, we set the missing lasts to the next 50 time points once it happens. We refer the 50 consecutive missing values to a missing window with length 50. Missing windows are randomly assigned without overlap to make 30 % of the whole time series are missing. GEFCom dataset is given with consecutive missing values. Each missing window consists of length 168 and 4 windows are included in the time series.

The forecast accuracy is represented by the Mean Squared Error (MSE) obtained on the unseen values of the test set. The lower MSE implies the higher forecast accuracy. In order to obtain a forecasting problem that is not too trivial, it is reasonable to select forecast time interval that guarantees to become linearly decorrelated. Hence, we consider the first zero of the autocorrelation function of the time series [4], 12 time steps ahead for Mackey-Glass (MG) system [9] and 24 time steps ahead for GEFCom 2012 dataset [11].

All RNNs are trained by truncated backpropagation though time, tBPTT(k_2, k_1) [4] with its tailored mini-batch formulation illustrated in Figure 5. Note that a chunk of tBPTT(k_2, k_1) have overlapped information of length $k_2 - k_1$ with neighboring chunks. This redundancy, obtained from the overlapped information, alleviates the impact that occurs in the drawback where the gradient is not fully backpropagated.



(a) MSE within missing windows

(b) MSE out of missing windows

(c) MSE of entire time series

Figure 6: MSE comparison among DRAN(5) and others with MG set.

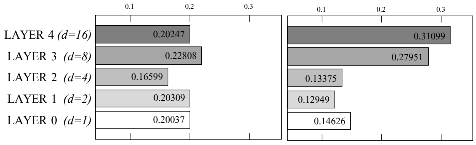
5 Results

Mackey-Glass Dataset

Figure 6 reports the forecast accuracy of MG test set with respect to MSE obtained from each model. To show the difference between the prediction performance of the different models with or without missing values in the input, the MSE presented in each subplot is computed on (a) within the missing windows; (b) out of the missing windows; and (c) entire time series.

In Figure 6(a), DRAN(5) outperforms other models with the lowest MSE(0.076), meanwhile, in Figure 6(b), DRNN(5) with missing mask outperforms other models with the lowest MSE(0.018). In Figure 6(c), DRNN(5) with binary mask outperforms other models with the lowest MSE(0.037) and DRAN(5) follows by 0.042.

An important sanity check for DRAN consists of observing the change of each attention weights $\{\alpha_t^{(l)}\}$ between when the input data is missing or not. We keep track of each weight and compare the change



(a) Mean of the weights out of missing windows

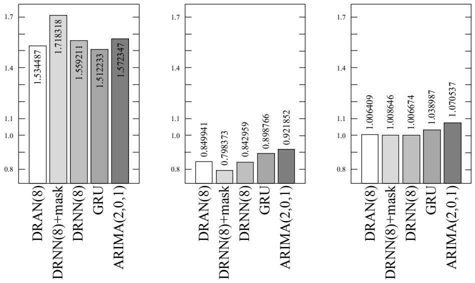
(b) Mean of the weights within missing windows

Figure 7: Comparison of the attention weights $\{\alpha_t^{(l)}\}$ *of DRAN(5) depending on input missingness with MG set.*

of mean values as attention weights play an indicating role revealing the layer that RNNs exploit. We argue that the change in the performance when the input data is missing or not supports the hypothesis that DRAN exploits the layer with the longer dilation more by redistributing finite attention resources when input value is consecutively missing. Figure 7(a) and (b) reveal that the average of attention weights of layer 3 ($d = 8$) and 4 ($d = 16$) strikingly increase while the weights of layer 1, 2 and 3 decrease within the missing windows, that supports the argument.

GEFCom Dataset

Figure 8 reports the forecast accuracy of GEFCom test set with respect to MSE in the same manner shown in Figure 6. In Figure 8(a), DRAN(8) results in the lowest MSE(1.534) among the dilated RNNs class, and second lowest MSE, followed by GRU(1.512) with small difference.

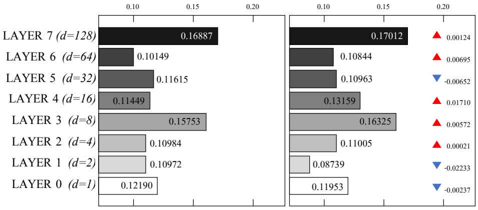


(a) MSE within missing windows

(b) MSE out of missing windows

(c) MSE of entire time series

Figure 8: MSE comparison among DRAN(8) and others with GEFCom 2012 set.



(a) Mean of the weights out of missing windows

(b) Mean of the weights within missing windows

Figure 9: Comparison of the attention weights $\{\alpha_t^{(l)}\}$ *of DRAN(8) depending on input missingness with GEFCom set.*

Figure 8(b), DRNN(8) with missing mask achieves the lowest MSE(0.798) and other DRNN-based models are followed by, DRNN(8)(0.843) and DRAN(8) (0.850). For the MSE of the entire time series shown in Figure 8 (c), DRNN-based models indicate similar MSE, achieving a lower MSE than two baselines.

The change between Figure 9(a) and (b) follows similar phenomenon in Figure 7 between two classes. The attention weights with dilation $d = \{64, 128\}$ increase, while others turn to decrease. It implies that DRAN(8) uses attention to find more reliable information on its own, although the attention mechanism has not shown a definite improvement in the forecasting performance.

6 Conclusion

In the paper, we propose a novel model DRAN(l) tailored for STLF tasks with missing data. The consistent results from the different datasets support that DRAN(l) learns how to capture the missingness and utilize multiple dilations to improve forecasting accuracy.

References

[1] E. Almeshaiei and H. Soltan. A methodology for electric power load forecasting. *Alexandria Engineering Journal*, 50(2):137–144, 2011.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

[3] F. M. Bianchi, E. De Santis, A. Rizzi, and A. Sadeghian. Short-term electric load forecasting using echo state networks and PCA decomposition. *IEEE Access*, 3:1931–1943, 2015.

[4] F. M. Bianchi, E. Maiorino, M. Kampffmeyer, A. Rizzi, and R. Jenssen. *Recurrent neural networks for short-term load forecasting: an overview and comparative analysis.* Springer, 2017.

[5] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. Hasegawa-Johnson, and T. Huang. Dilated recurrent neural networks. *NeurIPS*, 30:77–87, 2017.

[6] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, 8(1):6085, 2018.

[7] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Conference on Empirical Methods in Natural Language Processing*, 1:1724–1734, 2014.

[8] T.-H. Dang-Ha, F. M. Bianchi, and R. Olsson. Local short term electricity load forecasting: Automatic approaches. *International Joint Conference on Neural Networks*, 7:4267–4274, 2017.

[9] J. D. Farmer and J. J. Sidorowich. Predicting chaotic time series. *Physical Review Letter*, 59(8):845–848, 1987.

[10] T. Hong and M. Shahidehpour. Load forecasting case study. *EISPC, US Department of Energy*, 2015.

[11] Kaggle. GEFCom global energy forecasting competition, 2012.

[12] Z. C. Lipton, D. Kale, and R. Wetzel. Modeling missing data in clinical time series with RNNs. *Machine Learning for Healthcare Conference*, 56:253–270, 2016.

[13] I. Shpitser, K. Mohan, and J. Pearl. Missing data as a causal and probabilistic problem. *Conference on Uncertainty in Artificial Intelligence*, 31:802–811, 2015.

[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *Arxiv*, 2016.

[15] M. Woodward, W. Smith, and H. Tunstall-pedoe. Bias from missing values: sex differences in implication of failed venepuncture for the scottish heart health study. *International journal of epidemiology*, 20(2):379–383, 1991.

[16] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016.

14 | Paper 6

A Robustness Analysis of Personalized Propagation of Neural Prediction

Changkyu Choi, Michael Kampffmeyer, and Robert Jenssen

UiT The Arctic University of Norway

1 Introduction

Data without annotation are easy to obtain in the real-world, however, established supervised learning methods are not applicable to analyze them. Several learning approaches have been proposed in recent years to exploit the underlying structure of the data without requiring annotations [1, 2]. Semi-supervised learning aims to improve the predictive performance of these unsupervised approaches, by exploiting partially acquired annotations in the dataset. One recent promising line of work in this scheme makes use of graph neural networks (GNN) [3]. The data is expressed as a graph, where vertices are data samples and edges, given by an adjacency matrix $\mathbf{A}$, represent pairwise relationships between data points. Although these approaches achieve promising performance, they have so far been limited to applications, where the graph, in form of the adjacency matrix, is available. This is a severe limitation, as most available datasets do not include a predefined graph structure. To address this shortcoming, we investigate if the adjacency matrix $\mathbf{A}$ can be replaced with affinity matrices obtained directly from the data. As a first step into this direction, and in order to analyze its potential, we provide an analysis of how the current state-of-the-art semi-supervised approach, Personalized Propagation of Neural Predictions(PPNP)[4], is affected by changes in the affinity matrix.

2 Background

A popular concept of exploiting structured datasets is neighborhood aggregation, where large node neighborhoods are combined to achieve a more comprehensive representation. However, this often tends to cause over-smoothing and leads to a loss of the local structure in the neighborhood as the neighborhood size increases [5, 4]. To improve the over-smoothing issue commonly found in previous graph-based approaches, Klicpera et al.[4] suggest PPNP by adopting an idea from Personalized PageRank(PPR)[6]. Their model is given as,

$$\begin{aligned}
\mathbf{Z^{(0)}} &= \mathbf{H} = f_\theta(\mathbf{X}) \\
\mathbf{Z^{(k+1)}} &= (1-\alpha)\hat{\tilde{\mathbf{A}}}\mathbf{Z^{(k)}} + \alpha\mathbf{H} \\
\mathbf{Z^{(K)}} &= \sigma\Big((1-\alpha)\hat{\tilde{\mathbf{A}}}\mathbf{Z^{(K-1)}} + \alpha\mathbf{H}\Big)
\end{aligned} \tag{1}$$

where f_θ denotes a neural network, $\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^N$ is the network prediction, α is the teleport probablity, σ is the softmax, and $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ is an input feature matrix, where each data point is represented as a vertex in the graph. The adjacency matrix $\mathbf{A} = \Big\{a_{ij} \in \{0,1\}\Big\}_{i,j=1}^N$ represents the pairwise relationship of the points in $\mathbf{X}$ and $\hat{\tilde{\mathbf{A}}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ and $\tilde{\mathbf{D}}_{ii} = \Sigma_j \tilde{\mathbf{A}}_{ij}$. The main idea of PPR is to assign the restart state $\mathbf{h}_i$ for the node i and to aggregate neighborhood using the matrix $\hat{\tilde{\mathbf{A}}}$ with restart at any random propagation layer k. In addition, shown in Eq. (1), approximated PPNP has a separate two-step architecture with individual functionality; **(a)** neural network $f_\theta(\mathbf{X})$ which is related to the learning procedure; and **(b)** a K-layer propagation stack which exploits $\mathbf{A}$.

3 Methodology

For the robustness analysis of the PPNP framework, we define an ideal affinity matrix $\mathbf{A}_{ide}$ and analyze the effect of reducing the quality of the affinity matrix. We do this by replacing $\mathbf{A}$ in the framework with degenerative versions of $\mathbf{A}_{ide}$. This analysis aims to observe the change in accuracy with respect to the degree of degeneration.

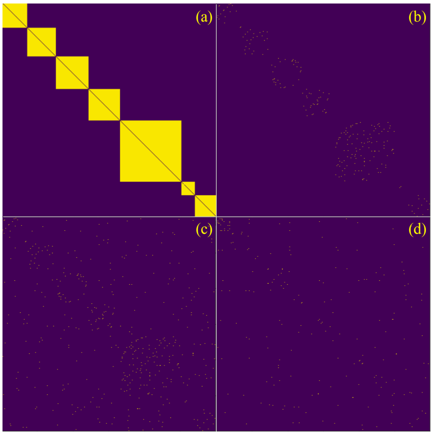


Figure 1: Degenerative versions of sorted $\mathbf{A}_{ide}$ of Cora-ML dataset, where the number of the classes $C = 7$, and the sample size in each class is $[354, 402, 452, 442, 857, 193, 295]$ (a) $\beta = \gamma = 0.00$ (b) $\beta = 0.99, \gamma = 0$ (c) $\beta = 0.99, \gamma = 2.32e^{-3}$ (d) observed/given graph $\mathbf{A}_{obs}$. Best viewed in electronic format (zoomed in).

Edge-reducing and Edge-activating Probabilities, β and γ: Let $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$ be a set of one-hot encoded label information for classification. We define the ideal matrix $\mathbf{A}_{ide} = \mathbf{Y}\mathbf{Y}^T - \mathbf{I}_N$.

Note that the ideal graph consists of several subgraphs where each of them represents one class as shown in Fig. 1(a). All nodes in a subgraph are fully connected to each other, meaning that for a node all other nodes in the class are the one-hop neighborhood. Meanwhile, nodes between different classes are disconnected.

Two variables, β and γ, are introduced to degrade the ideal graph. $\beta = \frac{r}{M}$ and $\gamma = \frac{t}{N_2 - N - M}$, where M is the total number of edges in $\mathbf{A}_{ide}$, r corresponds to the number of reduced edges ($0 \leq r \leq M$), and t is the number of activated edges ($0 \leq t \leq N^2 - N - M$). Edge-reducing probability β implies the removal of the edges in the graph of $\mathbf{A}_{ide}$ as shown in Fig. 1(b). It destroys the structure within a subgraph but, on the other hand, makes the matrix sparse and may increase the efficiency.

4 Analysis and Insight

Edge-activating probability γ implies the addition of edges between nodes in different classes. It causes the matrix do become more dense and noisy (see Fig. 1(c)).

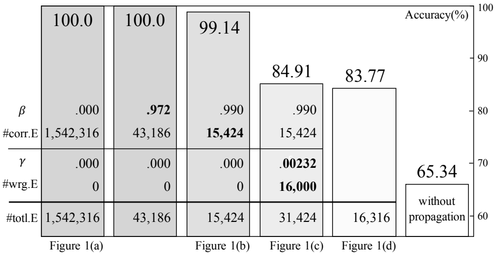


Figure 2: Result of the robustness analysis. Each bar-graph represents the accuracy from different affinity matrix.

To enable comparisons, hyperparameters match the original PPNP paper [4], including $\alpha = 0.1$ and $K = 10$ in Eq. (1). The Cora-ML benchmark dataset [3] is chosen for the analysis. Input feature matrix $\mathbf{X}$ has $N = 2,905$ datapoints with $D = 2,819$ features each and the observed adjacency matrix $\mathbf{A}_{obs}$ has 16,316 edges. By varying the β and γ parameters and performing extensive experiments, we observe among others, that perfect accuracy can be achieved even if 97.2% of the edges in $\mathbf{A}_{ide}$ are removed ($\beta = 0.972$) by reducing the "ideal" number of edges $(1, 542, 316)$ to $43, 186$ (see Figure 2). This is intuitive, as removing edges at random, still leaves the individual classes connected unless the graph is thinned too much. As long as there is a path that connects all nodes in the same class 100.0 % can be obtained. At the same time, this thinning reduces inference time approximately 15%. Further, the accuracy decreases as the number of wrong edges increase. Interestingly, thinning the graph to a similar size as the original Cora-ML dataset (by choosing $\beta = 0.99$) and doubling the number of edges by adding wrong edges ($\gamma = 0.0023$) still gives a performance of 84.91%. This is still more than the reported accuracy obtained by the PPNP approach, which is 83.77%.

5 Conclusion

We have analysed the state-of-the-art semi-supervised learning approach PPNP and provided insights into its robustness to the graph structure. This is done by replacing the adjacency matrix with degenerative versions of the ideal matrix $\mathbf{A}_{ide}$. In future work, we will extend this framework to semi-supervised problems without adjacency matrix to learn the network representations and the affinity matrix simultaneously.

References

[1] B. Huang, K. Zhang, P. Xie, M. Gong, E. P. Xing, and C. Glymour. Specific and shared causal relation modeling and mechanism-based clustering. *NeurIPS*, 2019.

[2] M. Kampffmeyer, S. Løkse, F. M. Bianchi, L. Livi, A.-B. Salberg, and R. Jenssen. Deep divergence-based approach to clustering. *Neural Networks*, 113:91–101, 2019.

[3] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

[4] J. Klicpera, A. Bojchevski, and S. Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. *ICLR*, 2019.

[5] Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *AAAI*, 2018.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Bibliography

[1] George R Cutter Jr and David A Demer. Accounting for scattering directivity and fish behaviour in multibeam-echosounder surveys. *ICES Journal of Marine Science*, 64(9):1664–1674, 2007.

[2] Rolf J Korneliussen and Egil Ona. Synthetic echograms generated from the relative frequency response. *ICES Journal of Marine Science*, 60(3): 636–640, 2003.

[3] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[5] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[6] 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*, 77(4):1391–1400, 2020.

[7] Guobao Xu, Weiming Shen, and Xianbin Wang. Applications of wireless sensor networks in marine environment monitoring: A survey. *Sensors*, 14(9):16932–16954, 2014.

[8] Rebecca Clausen and Brett Clark. The metabolic rift and marine ecology: An analysis of the ocean crisis within capitalist production. *Organization & environment*, 18(4):422–444, 2005.

[9] Steven A Murawski. Definitions of overfishing from an ecosystem perspective. *ICES Journal of Marine Science*, 57(3):649–658, 2000.

[10] Scott C Doney. The growing human footprint on coastal and open-ocean biogeochemistry. *science*, 328(5985):1512–1516, 2010.

[11] Ketil Malde, Nils Olav Handegard, Line Eikvil, and Arnt-Børre Salberg. Machine intelligence and the data-driven future of marine science. *ICES Journal of Marine Science*, 77(4):1274–1285, 2020.

[12] Carlos A Trasviña-Moreno, Rubén Blasco, Álvaro Marco, Roberto Casas, and Armando Trasviña-Castro. Unmanned aerial vehicle based wireless sensor network for marine-coastal environment monitoring. *Sensors*, 17 (3):460, 2017.

[13] Vaneeda Allken, Shale Rosen, Nils Olav Handegard, and Ketil Malde. A deep learning-based method to identify and count pelagic and mesopelagic fishes from trawl camera images. *ICES Journal of Marine Science*, 78(10):3780–3792, 2021.

[14] Pulkit Gupta, Jatin Batra, Jogender Sangwan, and Aanchal Khatri. Marine monitoring based on wsn: application and challenges. *International Journal of Advanced Studies of Scientific Research*, 3(12), 2018.

[15] Changkyu Choi, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, and Robert Jenssen. Deep semisupervised semantic segmentation in multifrequency echosounder data. *IEEE Journal of Oceanic Engineering*, 48(2):384–400, 2023. doi: 10.1109/JOE.2022.3226214.

[16] Smail Dilmi and Mohamed Ladjal. A novel approach for water quality classification based on the integration of deep learning and feature extraction techniques. *Chemometrics and Intelligent Laboratory Systems*, 214:104329, 2021.

[17] Archana Solanki, Himanshu Agrawal, and Kanchan Khare. Predictive analysis of water quality parameters using deep learning. *International Journal of Computer Applications*, 125(9):0975–8887, 2015.

[18] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

[19] IM Yusup, M Iqbal, and I Jaya. Real-time reef fishes identification using deep learning. In *IOP Conference Series: Earth and Environmental Science*, volume 429, page 012046. IOP Publishing, 2020.

[20] Alina Raphael, Zvy Dubinsky, David Iluz, Jennifer IC Benichou, and Nathan S Netanyahu. Deep neural network recognition of shallow water corals in the gulf of eilat (aqaba). *Scientific reports*, 10(1):12959, 2020.

[21] Rolf J Korneliussen. Acoustic target classification. *Coop. Res. Rep. - Int. Counc. Explor. Sea*, 344:104, 2018. doi: http://doi.org/10.17895/ices.pub.4567.

[22] John Simmonds and David N MacLennan. *Fisheries acoustics: theory and practice.* John Wiley & Sons, 2008.

[23] J Michael Jech, Matthias Schaber, Martin Cox, Pablo Escobar-Flores, Sven Gastauer, Kunnath Haris, John Horne, Toby Jarvis, Yoann Ladroit, Richard O'Driscoll, et al. Collecting quality echosounder data in inclement weather. 2021.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[25] Changkyu Choi, Michael Kampffmeyer, Nils Olav Handegard, Arnt-Børre Salberg, Olav Brautaset, Line Eikvil, and Robert Jenssen. Semi-supervised target classification in multi-frequency echosounder data. *ICES Journal of Marine Science*, 78(7):2615–2627, 2021.

[26] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[27] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[28] Wei Xiong, Zhenyu Xiong, and Yaqi Cui. An explainable attention network for fine-grained ship classification using remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022.

[29] Mobeen Nazar, Muhammad Mansoor Alam, Eiad Yafi, and Mazliham Mohd Su'ud. A systematic review of human–computer interaction and explainable artificial intelligence in healthcare with artificial intelligence techniques. *IEEE Access*, 9:153316–153348, 2021.

[30] Maria Sahakyan, Zeyar Aung, and Talal Rahwan. Explainable artificial intelligence for tabular data: A survey. *IEEE Access*, 9:135392–135422, 2021.

[31] Luca Longo, Randy Goebel, Freddy Lecue, Peter Kieseberg, and Andreas Holzinger. Explainable artificial intelligence: Concepts, applications, research challenges and visions. In *Machine Learning and Knowledge Extraction*, pages 1–16. Springer, 2020.

[32] David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI magazine*, 40(2):44–58, 2019.

[33] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[34] Espen Johnsen, Guillaume Rieucau, Egil Ona, and Georg Skaret. Collective structures anchor massive schools of lesser sandeel to the seabed, increasing vulnerability to fishery. *Marine Ecology Progress Series*, 573: 229–236, 2017.

[35] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.

[36] Shujian Yu, Luis Gonzalo Sanchez Giraldo, Robert Jenssen, and Jose C Principe. Multivariate extension of matrix-based rényi's α-order entropy functional. *IEEE transactions on pattern analysis and machine intelligence*, 42(11):2960–2966, 2019.

[37] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[38] Alfréd Rényi et al. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1. Berkeley, California, USA, 1961.

[39] Espen Johnsen, Ronald Pedersen, and Egil Ona. Size-dependent frequency response of sandeel schools. *ICES Journal of Marine Science*, 66 (6):1100–1105, 2009.

[40] Martin Biuw, Tor Arne Øigård, Kjell Tormod Nilssen, Garry Stenson, Lotta Lindblom, Michael Poltermann, Martin Kristiansen, and Tore Haug. Recent harp and hooded seal pup production estimates in the greenland sea suggest ecology-driven declines. *NAMMCO Sci. Publ.*, 12: 1–15, 2022.

[41] Kacper Sokol and Peter Flach. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 56–67, 2020.

[42] Holly Else. A guide to plan s: the open-access initiative shaking up science publishing. *Nature*, 2021.

[43] Carlos M Duarte, Susana Agusti, Edward Barbier, Gregory L Britten, Juan Carlos Castilla, Jean-Pierre Gattuso, Robinson W Fulweiler, Terry P Hughes, Nancy Knowlton, Catherine E Lovelock, et al. Rebuilding marine life. *Nature*, 580(7801):39–51, 2020.

[44] Egil Ona. An expanded target-strength relationship for herring. *ICES Journal of Marine Science*, 60(3):493–499, 2003.

[45] Nils Olav Handegard and Dag Tjøstheim. The sampling volume of trawl and acoustics: estimating availability probabilities from observations of tracked individual fish. *Canadian Journal of Fisheries and Aquatic Sciences*, 66(3):425–437, 2009.

[46] Kelly J Benoit-Bird and Gareth L Lawson. Ecological insights from pelagic habitats acquired using active acoustic techniques. *Annual review of marine science*, 8:463–490, 2016.

[47] RM Levine, A De Robertis, D Grünbaum, and CD Wilson. Transport-driven seasonal abundance of pelagic fishes in the chukchi sea observed with seafloor-mounted echosounders. *ICES Journal of Marine Science*, page fsad024, 2023.

[48] Kenneth G Foote. Linearity of fisheries acoustics, with addition theorems. *The Journal of the Acoustical Society of America*, 73(6):1932–1940, 1983.

[49] Rolf J Korneliussen, Noel Diner, Egil Ona, Laurent Berger, and Paul G Fernandes. Proposals for the collection of multifrequency acoustic data. *ICES Journal of Marine Science*, 65(6):982–994, 2008.

[50] Mikael van Deurs, Asbjørn Christensen, and Anna Rindorf. Patchy zooplankton grazing and high energy conversion efficiency: ecological implications of sandeel behavior and strategy. *Marine Ecology Progress Series*, 487:123–133, 2013.

[51] Anna Rindorf, Peter J Wright, Henrik Jensen, and Marie Maar. Spatial differences in growth of lesser sandeel in the north sea. *Journal of Experimental Marine Biology and Ecology*, 479:9–19, 2016.

[52] Alan MacDonald, Michael R Heath, Simon PR Greenstreet, and Douglas C Speirs. Timing of sandeel spawning and hatching off the east coast of scotland. *Frontiers in Marine Science*, 6:70, 2019.

[53] Rokas Kubilius and Egil Ona. Target strength and tilt-angle distribution of the lesser sandeel (ammodytes marinus). *ICES Journal of Marine Science*, 69(6):1099–1107, 2012.

[54] Zheng Zeng, Lian Lian, Karl Sammut, Fangpo He, Youhong Tang, and Andrew Lammas. A survey on path planning for persistent autonomy of autonomous underwater vehicles. *Ocean Engineering*, 110:303–313, 2015.

[55] Oscar Sund. Echo sounding in fishery research. *Nature*, 135(3423):953–953, 1935.

[56] Godfrey M Hewitt. Genetic consequences of climatic oscillations in the quaternary. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 359(1442):183–195, 2004.

[57] JL Watkins, K Reid, D Ramm, XY Zhao, M Cox, G Skaret, S Fielding, XL Wang, and E Niklitschek. The use of fishing vessels to provide acoustic data on the distribution and abundance of antarctic krill and other pelagic species. *Fisheries Research*, 178:93–100, 2016.

[58] Lavinia Suberg, Russell B Wynn, Jeroen Van Der Kooij, Liam Fernand, Sophie Fielding, Damien Guihen, Douglas Gillespie, Mark Johnson, Kalliopi C Gkikopoulou, Ian J Allan, et al. Assessing the potential of autonomous submarine gliders for ecosystem monitoring across multiple trophic levels (plankton to cetaceans) and pollutants in shallow shelf seas. *Methods in Oceanography*, 10:70–89, 2014.

[59] Charles H Greene, Erin L Meyer-Gutbrod, Louise P McGarry, Lawrence C Hufnagle Jr, Dezhang Chu, Sam McClatchie, Asa Packer, Jae-Byung Jung, Timothy Acker, Huck Dorn, et al. A wave glider approach to fisheries acoustics: transforming how we monitor the nation's commercial fisheries in the 21st century. *Oceanography*, 27(4):168–174, 2014.

[60] Rolf J Korneliussen and Egil Ona. An operational system for processing and visualizing multi-frequency acoustic data. *ICES Journal of Marine Science*, 59(2):293–313, 2002.

[61] Rolf J Korneliussen. Measurement and removal of echo integration noise. *ICES Journal of Marine Science*, 57(4):1204–1217, 2000.

[62] Suchita Nanaware, Rajveer Shastri, Yashwant Joshi, and Arnab Das. Passive acoustic detection and classification of marine mammal vocalizations. In *2014 International Conference on Communication and Signal Processing*, pages 493–497. IEEE, 2014.

[63] Zhen Ye. Resonant scattering of acoustic waves by ellipsoid air bubbles in liquids. *The Journal of the Acoustical Society of America*, 101(2): 681–685, 1997.

[64] RJ Kloser, T Ryan, P Sakov, A Williams, and JA Koslow. Species identification in deep water using multiple acoustic frequencies. *Canadian Journal of Fisheries and Aquatic Sciences*, 59(6):1065–1077, 2002.

[65] Laura Mannocci, Yannick Baidai, Fabien Forget, Mariana Travassos Tolotti, Laurent Dagorn, and Manuela Capello. Machine learning to detect bycatch risk: Novel application to echosounder buoys data in tuna purse seine fisheries. *Biological Conservation*, 255:109004, 2021.

[66] Karolina Trzcinska, Lukasz Janowski, Jaroslaw Nowak, Maria Rucinska-Zjadacz, Aleksandra Kruss, Jens Schneider von Deimling, Pawel Pocwiardowski, and Jaroslaw Tegowski. Spectral features of dual-frequency multibeam echosounder data for benthic habitat mapping. *Marine Geology*, 427:106239, 2020.

[67] Mirjam Snellen, Kerstin Siemes, and Dick G Simons. Model-based sediment classification using single-beam echosounder signals. *The Journal of the Acoustical Society of America*, 129(5):2878–2888, 2011.

[68] Jon Lopez, Gala Moreno, Guillermo Boyra, and Laurent Dagorn. A model based on data from echosounder buoys to estimate biomass of fish

species associated with fish aggregating devices. *Fishery Bulletin*, 114 (2):166–178, 2016.

[69] Rolf J Korneliussen, Yngve Heggelund, Gavin J Macaulay, Daniel Patel, Espen Johnsen, and Inge K Eliassen. Acoustic identification of marine species using a feature library. *Methods in Oceanography*, 17:187–205, 2016.

[70] G. A. Rose and W. C. Leggett. Hydroacoustic signal classification of fish schools by species. *Canadian Journal of Fisheries and Aquatic Sciences*, 45(4):597–604, 1988. doi: 10.1139/f88-073.

[71] William A Karp and Gary E Walters. Survey assessment of semi-pelagic gadoids: the example of walleye pollock, theragra chalcogramma, in the eastern bering sea. 1994.

[72] Sven Gastauer, Sascha MM Fässler, Ciaran O'Donnell, Åge Høines, Jan Arge Jakobsen, Alexander I Krysov, Leon Smith, Øyvind Tangen, Valantine Anthonypillai, Ebba Mortensen, et al. The distribution of blue whiting west of the british isles and ireland. *Fisheries Research*, 183:32–43, 2016.

[73] Qingxin Meng, Shie Yang, and Shengchun Piao. The classification of underwater acoustic target signals based on wave structure and support vector machine. *The Journal of the Acoustical Society of America*, 136 (4):2265–2265, 2014.

[74] Wen Zhang, Yanqun Wu, Dezhi Wang, Yongxian Wang, Yibo Wang, and Lilun Zhang. Underwater target feature extraction and classification based on gammatone filter and machine learning. In *2018 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pages 42–47. IEEE, 2018.

[75] Bart Buelens, Tim Pauly, Raymond Williams, and Arthur Sale. Kernel methods for the detection and classification of fish schools in single-beam and multibeam acoustic data. *ICES Journal of Marine Science*, 66(6): 1130–1135, 02 2009.

[76] BM Sherin and MH Supriya. Selection and parameter optimization of svm kernel function for underwater target classification. In *2015 IEEE Underwater Technology (UT)*, pages 1–5. IEEE, 2015.

[77] Van-Sang Doan, Thien Huynh-The, and Dong-Seong Kim. Underwater acoustic target classification based on dense convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2020.

[78] Shengzhao Tian, Duanbing Chen, Hang Wang, and Jingfa Liu. Deep convolution stack for waveform in underwater acoustic target recognition. *Scientific reports*, 11(1):9614, 2021.

[79] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[80] Dinh Quang Huy, Nicholas Sadjoli, Abu Bakr Azam, Basman Elhadidi, Yiyu Cai, and Gerald Seet. Object perception in underwater environments: a survey on sensors and sensing methodologies. *Ocean Engineering*, 267:113202, 2023.

[81] R Glenn Wright. Intelligent autonomous ship navigation using multi-sensor modalities. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, 13(3), 2019.

[82] Fahimeh Farahnakian and Jukka Heikkonen. Deep learning based multi-modal fusion architectures for maritime vessel detection. *Remote Sensing*, 12(16):2509, 2020.

[83] Minsung Sung, Jason Kim, Hyeonwoo Cho, Meungsuk Lee, and Son-Cheol Yu. Underwater-sonar-image-based 3d point cloud reconstruction for high data utilization and object classification using a neural network. *Electronics*, 9(11):1763, 2020.

[84] ICES. Report of the benchmark on sandeel (wksand 2016). *ICES CM 2016/ACOM*, 33:1–319, 2017.

[85] CT Macer. Sand eels (ammodytidae) in the south-western north sea; their biology and fishery. 1966.

[86] PJ Wright, Henrik Jensen, and I Tuck. The influence of sediment type on the distribution of the lesser sandeel, ammodytes marinus. *Journal of Sea Research*, 44(3-4):243–256, 2000.

[87] Robert W Furness. Management implications of interactions between fisheries and sandeel-dependent seabirds and seals in the north sea. *ICES Journal of Marine Science*, 59(2):261–269, 2002.

[88] Verena M Trenkel, Valérie Mazauric, and Laurent Berger. The new fisheries multibeam echosounder me70: description and expected contribution to fisheries research. *ICES Journal of Marine Science*, 65(4):645–655, 2008.

[89] David N MacLennan and E John Simmonds. *Fisheries acoustics*, volume 5. Springer Science & Business Media, 2013.

[90] GB Stenson, L-P Rivest, MO Hammill, and JF Gosselin. Estimating pup production of harp seals, pagophilus groenlandicus, in the northwest atlantic. *Mar. Mamm. Sci.*, 19(1):141–160, 2003.

[91] VA Potelov, AP Golikov, and VA Bondarev. Estimated pup production of harp seals pagophilus groenlandicus in the white sea, russia, in 2000. *ICES J. Mar. Sci.*, 60(5):1012–1017, 2003.

[92] Michael O Hammill, CE den Heyer, and WD Bowen. Grey seal population trends in canadian waters, 1960-2014. *DFO Can. Sci. Advis. Sec. Res. Doc.*, pages 1–44, 2014.

[93] Kristoffer Wickstrøm, Michael Kampffmeyer, and Robert Jenssen. Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps. *Medical image analysis*, 60: 101619, 2020.

[94] Stine Hansen, Srishti Gautam, Robert Jenssen, and Michael Kampffmeyer. Anomaly detection-inspired few-shot medical image segmentation through self-supervision with supervoxels. *Medical Image Analysis*, 78:102385, 2022.

[95] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2022.

[96] Qinghui Liu, Michael Kampffmeyer, Robert Jenssen, and Arnt-Børre Salberg. Multi-modal land cover mapping of remote sensing images using pyramid attention and gated fusion networks. *International Journal of Remote Sensing*, 43(9):3509–3535, 2022.

[97] Luigi Tommaso Luppino, Mads Adrian Hansen, Michael Kampffmeyer, Filippo Maria Bianchi, Gabriele Moser, Robert Jenssen, and Stian Normann Anfinsen. Code-aligned autoencoders for unsupervised change detection in multimodal remote sensing images. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[98] Sara Björk, Stian Normann Anfinsen, Erik Naesset, Terje Gobakken, and Eliakimu Zahabu. On the potential of sequential and nonsequential

regression models for sentinel-1-based biomass prediction in tanzanian miombo forests. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:4612–4639, 2022.

[99] Alba Ordoñez, Line Eikvil, Arnt-Børre Salberg, Alf Harbitz, Sean Melling Murray, and Michael C Kampffmeyer. Explaining decisions of deep neural networks used for fish age prediction. *PloS one*, 15(6):e0235013, 2020.

[100] Alba Ordoñez, Ingrid Utseth, Olav Brautaset, Rolf Korneliussen, and Nils Olav Handegard. Evaluation of echosounder data preparation strategies for modern machine learning models. *Fisheries Research*, 254: 106411, 2022.

[101] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[102] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[103] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern recognition*. Elsevier, 2006.

[104] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

[105] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[106] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[107] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[108] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[109] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[110] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[111] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[112] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, (ICLR)*, 2021.

[113] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[114] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5028–5037, 2017.

[115] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[116] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005.

[117] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[118] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[119] Larry Biehl and David Landgrebe. Multispec—a tool for multispectral–hyperspectral image data analysis. *Computers & Geosciences*, 28(10): 1153–1159, 2002.

[120] Olivier Debeir, Isabelle Van den Steen, Patrice Latinne, Phlllppe Van Ham, and Eleonore Wolff. Textural and contextual land-cover classification using single and multiple classifier systems. *Photogrammetric Engineering and Remote Sensing*, 68(6):597–606, 2002.

[121] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, (ICLR)*, 2015.

[122] 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.

[123] Dhruv Rathi, Sushant Jain, and S Indu. Underwater fish species classification using convolutional neural network and deep learning. In *2017 Ninth international conference on advances in pattern recognition (ICAPR)*, pages 1–6. Ieee, 2017.

[124] Thomas Haugland Johansen, Steffen Aagaard Sørensen, Kajsa Møllersen, and Fred Godtliebsen. Instance segmentation of microscopic foraminifera. *Applied Sciences*, 11(14):6543, 2021.

[125] Rajendra Sapkota, Puneet Sharma, and Ingrid Mann. Comparison of deep learning models for the classification of noctilucent cloud images. *Remote Sensing*, 14(10):2306, 2022.

[126] Andreas Kvammen, Kristoffer Wickstrøm, Derek McKay, and Noora Partamies. Auroral image classification with deep neural networks. *Journal of Geophysical Research: Space Physics*, 125(10): e2020JA027808, 2020.

[127] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.

[128] Fahad Lateef and Yassine Ruichek. Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, 338:321–348, 2019.

[129] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.*, 22(3):1341–1360, 2020.

[130] Stine Hansen, Samuel Kuttner, Michael Kampffmeyer, Tom-Vegard Markussen, Rune Sundset, Silje Kjærnes Øen, Live Eikenes, and Robert Jenssen. Unsupervised supervoxel-based lung tumor segmentation across patient scans in hybrid pet/mri. *Expert Syst. Appl*, 167:114244, 2021.

[131] Mohamed A. Naser and M. Jamal Deen. Brain tumor segmentation and grading of lower-grade glioma using deep learning in mri images. *Comput. Biol. Med.*, 121:103758, 2020.

[132] Debesh Jha, Sharib Ali, Nikhil Kumar Tomar, Håvard D Johansen, Dag Johansen, Jens Rittscher, Michael A Riegler, and Pål Halvorsen. Real-time polyp detection, localization and segmentation in colonoscopy using deep learning. *IEEE Access*, 9:40496–40510, 2021.

[133] Qinghui Liu, Michael Kampffmeyer, Robert Jenssen, and Arnt-Børre Salberg. Dense dilated convolutions' merging network for land cover classification. *IEEE Trans. Geosci. Remote Sens.*, 58(9):6309–6320, 2020.

[134] Ava Vali, Sara Comai, and Matteo Matteucci. Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review. *Remote Sens.*, 12(15):2495, 2020.

[135] Luigi Tommaso Luppino, Michael Kampffmeyer, Filippo Maria Bianchi, Gabriele Moser, Sebastiano Bruno Serpico, Robert Jenssen, and Stian Normann Anfinsen. Deep image translation with an affinity-based change prior for unsupervised multimodal change detection. *IEEE Trans. Geosci. Remote Sens.*, 2021.

[136] Xiao Yu, Junfu Fan, Jiahao Chen, Peng Zhang, Yuke Zhou, and Liusheng Han. NestNet: a multiscale convolutional neural network for remote sensing image change detection. *Int. J. Remote Sens.*, 42(13):4898–4921, 2021.

[137] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[138] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB)*, pages 1–7. IEEE, 2020.

[139] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015.

[140] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[141] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[142] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: DLMIA, ML-CDS 2017, Held in Conjunction with MICCAI*, pages 240–248. Springer, 2017.

[143] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[144] Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba Amina Salahuddin, Robert Jenssen, Marina Höhne, and Michael Kampffmeyer. Protovae: A trustworthy self-explainable prototypical variational model. *Advances in neural information processing systems*, 2022.

[145] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *Machine Learning in Medical Imaging: 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings 8*, pages 379–387. Springer, 2017.

[146] Seyed Raein Hashemi, Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, Sanjay P Prabhu, Simon K Warfield, and Ali Gholipour. Asymmetric loss functions and deep densely-connected networks for highly-imbalanced

medical image segmentation: Application to multiple sclerosis lesion detection. *IEEE Access*, 7:1721–1735, 2018.

[147] Javier Ribera, David Guera, Yuhao Chen, and Edward J Delp. Locating objects without bounding boxes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6489, 2019.

[148] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[149] William Bialek Naftali Tishby, Fernando C.Pereira. The information bottleneck method. In *Proceedings of the Thirty-Seventh Annual Allerton Conference on Communication, Control and Computing*, pages 368—-377, 1999.

[150] Tatiana Ermakova, Julia Blume, Benjamin Fabian, Elena Fomenko, Marcus Berlin, and Manfred Hauswirth. Beyond the hype: why do data-driven projects fail? In *Proceedings of the 54th Hawaii International Conference on System Sciences*, page 5081, 2021.

[151] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145:90–106, 2022.

[152] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[153] Xiao Zhang, Yixiao Ge, Yu Qiao, and Hongsheng Li. Refining pseudo labels with clustering consensus over generations for unsupervised object re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3436–3445, 2021.

[154] Zhu Xiaojin. Semi-supervised learning literature survey. *Computer Sciences TR*, 1530, 2008.

[155] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[156] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020.

[157] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017.

[158] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6728–6736, 2019.

[159] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[160] Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. *arXiv preprint arXiv:1804.09530*, 2018.

[161] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Proc. Int. Conf. Mach. Learn. (ICML)*, 3(2):896, 2013.

[162] Hyeonsoo Lee and Won-Ki Jeong. Scribble2label: Scribble-supervised cell segmentation via self-generating pseudo-labels with consistency. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*, pages 14–23. Springer, 2020.

[163] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[164] Wenqiao Zhang, Lei Zhu, James Hallinan, Shengyu Zhang, Andrew Makmur, Qingpeng Cai, and Beng Chin Ooi. Boostmis: Boosting medical image semi-supervised learning with adaptive pseudo labeling and informative active annotation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20666–20676, 2022.

[165] Joo-Kyung Kim and Young-Bum Kim. Pseudo labeling and negative feedback learning for large-scale multi-label domain classification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7964–7968. IEEE, 2020.

[166] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alexey Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*, 2022.

[167] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2021.

[168] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4248–4257, 2022.

[169] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 299–315, 2018.

[170] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5070–5079, 2019.

[171] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11557–11568, 2021.

[172] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[173] Siti Nurulain Mohd Rum and Fariz Az Zuhri Nawawi. Fishdetec: A fish identification application using image recognition approach. *International Journal of Advanced Computer Science and Applications*, 12(3), 2021.

[174] Alba Ordoñez, Line Eikvil, Arnt-Børre Salberg, Alf Harbitz, and Bjarki Þór Elvarsson. Automatic fish age determination across different otolith image labs using domain adaptation. *Fishes*, 7(2):71, 2022.

[175] Alessandra Lumini, Loris Nanni, and Gianluca Maguolo. Deep learning for plankton and coral classification. *Applied Computing and Informatics*, 2020.

[176] Yassine Himeur, Bhagawat Rimal, Abhishek Tiwary, and Abbes Amira. Using artificial intelligence and data fusion for environmental monitoring: A review and future perspectives. *Information Fusion*, 2022.

[177] Ioannis Kakogeorgiou and Konstantinos Karantzalos. Evaluating explainable artificial intelligence methods for multi-label deep learning classification tasks in remote sensing. *International Journal of Applied Earth Observation and Geoinformation*, 103:102520, 2021.

[178] Xiao Bai, Xiang Wang, Xianglong Liu, Qiang Liu, Jingkuan Song, Nicu Sebe, and Been Kim. Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments. *Pattern Recognition*, 120:108102, 2021.

[179] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[180] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[181] Ian Covert, Scott M Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *J. Mach. Learn. Res.*, 22: 209–1, 2021.

[182] Gil Fidel, Ron Bitton, and Asaf Shabtai. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020.

[183] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

[184] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[185] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR*, 2015.

[186] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[187] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[188] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[189] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.

[190] Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*, 150:228–234, 2021.

[191] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference, BMVC*, page 151. BMVA Press, 2018.

[192] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[193] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[194] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *Advances in neural information processing systems*, 30, 2017.

[195] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[196] Seojin Bang, Pengtao Xie, Heewook Lee, Wei Wu, and Eric Xing. Explaining a black-box by using a deep variational information bottleneck approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11396–11404, 2021.

[197] Andreas Holzinger. From machine learning to explainable ai. In *2018 world symposium on digital intelligence for systems and machines (DISA)*, pages 55–66. IEEE, 2018.

[198] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[199] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[200] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces*, pages 126–137, 2015.

[201] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[202] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[203] Q Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the ai: informing design practices for explainable ai user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2020.

[204] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

[205] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

[206] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations, ICLR*, 2019.

[207] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR*, 2017.

[208] Jose C Principe. *Information theoretic learning: Renyi's entropy and kernel perspectives.* Springer Science & Business Media, 2010.

[209] Hanjie Chen, Faeze Brahman, Xiang Ren, Yangfeng Ji, Yejin Choi, and Swabha Swayamdipta. Information-theoretic evaluation of free-text rationales with conditional $\mathcal{V}$-information. In *NeurIPS Workshop on Trustworthy and Socially Responsible Machine Learning*, 2022.

[210] Changhee Lee and Mihaela van der Schaar. A variational information bottleneck approach to multi-omics data integration. In *International Conference on Artificial Intelligence and Statistics*, pages 1513–1521. PMLR, 2021.

[211] Inaki Estella Aguerri and Abdellatif Zaidi. Distributed variational representation learning. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):120–138, 2019.

[212] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck. In *International Conference on Learning Representations*, 2020.

[213] Ahcène Boubekki, Michael Kampffmeyer, Ulf Brefeld, and Robert Jenssen. Joint optimization of an autoencoder for clustering and embedding. *Machine Learning*, 110(7):1901–1937, 2021.