Automatic detection and classification of beluga whale calls in the St. Lawrence estuary

Article *in* The Journal of the Acoustical Society of America · December 2024

DOI: 10.1121/10.0030472

11 authors, including:

Jaclyn Aubin
Memorial University of Newfoundland
7 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Marie-Ana Mikus
Raincoast Conservation Foundation
8 PUBLICATIONS **109** CITATIONS
SEE PROFILE

Valeria Vergara
Raincoast Conservation Foundation
29 PUBLICATIONS **425** CITATIONS
SEE PROFILE

Sébastien Gambs
Université du Québec à Montréal
138 PUBLICATIONS **2,814** CITATIONS
SEE PROFILE

JASA ARTICLE

Automatic detection and classification of beluga whale calls in the St. Lawrence estuary

Tristan Cotillard,[1,2,a)] Xavier Sécheresse,[1,2,b)] Jaclyn Aubin,[3] Marie-Ana Mikus,[3] Valeria Vergara,[3] Sébastien Gambs,[4] Robert Michaud,[5] Cristiane C. A. Martins,[6] Samuel Turgeon,[6] Clément Chion,[1,c)] and Irene Roca[1,d)]

[1]*Department of Natural Sciences, Université du Quebec en Outaouais, Gatineau, Quebec, Canada*

[2]*Mines Paris, Paris Sciences et Lettres University, Paris, France*

[3]*Raincoast Conservation Foundation, Victoria, British Columbia, Canada*

[4]*Université du Québec à Montréal, Montréal, Quebec, Canada*

[5]*Groupe de Recherche et d'Éducation sur les Mammifères Marins, Tadoussac, Quebec, Canada*

[6]*Parks Canada, Gatineau, Quebec, Canada*

ABSTRACT:

The endangered beluga whale (*Delphinapterus leucas*) of the St. Lawrence Estuary (SLEB) faces threats from a variety of anthropogenic factors. Since belugas are a highly social and vocal species, passive acoustic monitoring has the potential to deliver, in a non-invasive and continuous way, real-time information on SLEB spatiotemporal habitat use, which is crucial for their monitoring and conservation. In this study, we introduce an automatic pipeline to analyze continuous passive acoustic data and provide standard and accurate estimations of SLEB acoustic presence and vocal activity. An object detector extracted vocalizations of beluga whales from an acoustic recording of beluga vocal activity. Then, two deep learning classifiers discriminated between high-frequency call types (40–120 kHz) and the presence of low-frequency components (0–20 kHz), respectively. Different algorithms were tested for each step and their main combinations were compared in time and performance. We focused our work on a high residency area, Baie Sainte-Marguerite (BSM), used for socialization and feeding by SLEB. Overall, this project showed that accurate continuous analysis of SLEB vocal activity at BSM could provide valuable information to estimate habitat use, link beluga behavior and acoustic activity within and between herds, and quantify beluga presence and abundance.

 https://doi.org/10.1121/10.0030472

(Received 4 June 2024; revised 16 September 2024; accepted 20 September 2024; published online 5 December 2024)

[Editor: Shane Guan] Pages: 3723–3740

I. INTRODUCTION

The beluga whale (*Delphinapterus leucas*) population inhabiting the waters of the St. Lawrence Estuary (SLEB) is considered endangered under the Canadian Species at Risk Act [Committee on the Status of Endangered Wildlife in Canada (COSEWIC), 2014] (Goverment of Canada, 2014). Marine traffic, noise pollution, contaminants, and a decline in prey abundance are among the main threats to their population (Fisheries and Oceans Canada, 2012). Yet, the magnitude of their impacts on beluga spatiotemporal habitat use and population dynamics remain mostly unknown. Continuously monitoring SLEB presence and spatiotemporal dynamics, potentially in real-time, could greatly enhance our understanding of these effects.

Several methods can be used to study mammals' activity and among them, passive acoustic monitoring (PAM) has proven very useful to monitor in a non-invasive and continuous way the spatiotemporal habitat use of marine mammals, especially threatened and endangered species inhabiting remote and difficult of access regions (Castellote *et al.*, 2020; Oedekoven *et al.*, 2022; Todd *et al.*, 2020). Contrary to visual surveys (land-based, by boat, drone, plane, or satellite), PAM allows continuous data collection at a wide range of spatial scales, independently of weather conditions or time of the day, and at different depths. However, the applicability of PAM depends on the patterns and consistency of the vocal behavior of the studied species. In addition, PAM generates a substantial volume of data, impractical to manage manually to provide relevant and up-to-date information that could be used to support dynamic conservation plans. Furthermore, manual protocols are not standardized and rely heavily on human effort (Kowarski and Moors–Murphy, 2021).

[a)]Email: tristan.cotillard@etu.minesparis.psl.eu
[b)]Email: xavier.secheresse@etu.minesparis.psl.eu
[c)]Email: clement.chion@uqo.ca
[d)]Email: irene.rocatorrecilla@uqo.ca

Belugas rely heavily on acoustic signals to communicate, navigate, feed, and socialize (Chmelnitsky and Ferguson, 2012; Lesage *et al.*, 1999; Panova *et al.*, 2012). The vocal repertoire of SLEB comprises broadband calls, such as contact calls and high-frequency pulsed calls, low-frequency modulated or pulsed tones, mixed calls that combine pulses and whistles, and echolocation clicks (Fish and Mowbray, 1962; Karlsen *et al.*, 2002; Sjare and Smith, 1986; Vergara and Mikus, 2018). Through the automatic and remote detection of single calls and their categorization (from broad to more specific classes) it is possible to analyze the rate and density of such acoustic categories. This makes it possible to infer the presence and spatiotemporal dynamics of SLEBs, and potentially to estimate abundance (Simard *et al.*, 2010), depict predominant behaviors within the herd [i.e., feeding, socially interacting, etc. (Castellote *et al.*, 2020)], infer group composition (Vergara *et al.*, 2021), and potentially recognize individual or herd-related acoustic signatures of interest (Panova *et al.*, 2012; Vergara and Mikus, 2018).

The use of machine learning to analyze PAM data has made it possible to automate the detection and classification of acoustic signals of interest and to extend the scale of the spatiotemporal analysis. More specifically, deep learning techniques, such as convolutional neural networks (CNNs), have emerged as state-of-the-art methods for target sound detection, species classification, and call type recognition (Allen *et al.*, 2021; Bergler *et al.*, 2022; Zhong *et al.*, 2020). Very recently, a more general idea using transformers was developed with AVES (Animal Vocalization Encoder based on Self-Supervision) (Hagiwara, 2022). AVES is designed to directly encode bioacoustic recordings into spectral features, which can then be used to distinguish acoustic classes (e.g., call types, species, etc.) through machine learning algorithms. This method yields promising results but has not been applied to marine mammal species nor to acoustic signals with very high and wide frequency bandwidth components ($\geq$100 kHz) yet. To the best of our knowledge, no previous work has developed an integrative workflow able to detect, accurately locate in time, and classify beluga whale call types.

The aim of this paper is to present a fully automatic pipeline, based on the flexible use and combination of several machine learning algorithms (i.e., convolutional neural networks and transformer models), and its applicability to the detection and classification of SLEB vocalizations from continuous long-term acoustic recordings.

II. MATERIALS AND METHODS

The main idea of our method is to separate long audio clips into fixed time fragments (order of seconds) and successively apply a detection algorithm, a high-frequency call classifier algorithm, and a low-frequency call detector algorithm on detected calls. For detection, the workflow includes two alternative options, a double thresholding technique allowing one to delimit regions of interest (ROI) and a detection transformer DETR (Carion *et al.*, 2020). For classification we tested convolutional residual networks (ResNet) and AVES transformer encoder fine-tuned. All algorithms were trained and tested to either detect or classify four SLEB's vocalization types, including contact calls (CC), high-frequency pulsed calls (HFPC), low-frequency narrow-band tonal signals (hereinafter referred to as whistles), and echolocation clicks (EC). The workflow allows a flexible combination of detection and classification algorithms to optimize the output depending on the scientific objective and the computational speed.

A. Data

Acoustic recordings were collected with a SoundTrap HF300 hydrophone (Ocean Instruments, NZ) deployed at the entrance of Baie Sainte Marguerite (BSM) (long- = −69.967525, lat = 48.24992). The hydrophone was fastened to an observation tower just above the river bottom at a depth of 15 to 20 m. It recorded continuously for 544 h and 38 min in 2017 and 824 h and 18 min in 2018, for a total of 57 days, using a 288 kHz sampling rate and 16 bit depth. The recorder had a flat frequency response over the 0.02–150 kHz (±3 dB) range and –172.7 dB re 1 V/μPa sensitivity.

We assessed the acoustic presence/absence of SLEB and evaluated the vocalization categories by visually and aurally identifying all SLEB signals on spectrograms of the acoustic recordings (with custom scripts). Four main call types known to be related to social and feeding behaviors were considered in the study in order to automate their detection and classification (Fig. 1). (i) CC included simple and complex calls. Simple CC are fast, broadband pulse trains with high acoustic energy up to 150 kHz (>0.6 s duration) and are used for group cohesion and mother-calf contact (Vergara *et al.*, 2010). Complex CC have a simultaneous low-frequency component superimposed on the main pulse train. This whistle-like component is suspected to encode individual information (Vergara and Mikus, 2018). (ii) High-frequency pulsed calls (HFPC) are also broadband calls starting at about 40 kHz. We also included simple and complex calls, the latter with a simultaneous low-frequency component. HFPC can be distinguished from burst pulses in echolocation buzz trains because they maintain a high and constant pulse repetition rate throughout the signal and because they are not immediately preceded or followed by echolocation clicks. (iii) EC range from about 30 to 120 kHz or higher (Au *et al.*, 1985; Song *et al.*, 2023) and are used to locate food and navigate (Panova *et al.*, 2012; Sjare and Smith, 1986). Note that we did not aim to detect and classify individual echolocation pulses, but rather, EC refers to echolocation events. (iv) The whistle category comprised all low-frequency (from about 500 Hz to 20 kHz) modulated narrow-band tonal or pulsed-tonal signals. Whistles are frequently produced during social interactions and directional swimming (Belikov and Bel'kovich, 2003; Garland *et al.*, 2015; Panova *et al.*, 2012),

https://doi.org/10.1121/10.0030472

and although their specific function is not yet well known, their detection also extends the spatial range over which SLEB can be detected, since low-frequency components are the ones that travel further in the water [e.g., 3–5 kHz (Vergara *et al.*, 2021)] (see Fig. 1 for spectrogram examples of these call types).

Only complete calls, with signal-to-noise ratios (SNR) ≥ 1 dB and clearly assigned to the abovementioned categories by expert bioacousticians with experience analyzing beluga calls were labeled. The SNR was computed following the implementation given by scikit-maad library (Ulloa *et al.*, 2021) (https://scikit-maad.github.io/generated/maad.sound.temporal_snr.html). We set up three different datasets to train and test the performance of the workflow in detecting and classifying SLEB call types. The first, or call dataset, comprised a total of 14 329 snippets of varying length, cut according to the duration of the labeled calls (ranging from 0.4 to 3 s). Five classes were labeled: CC (n = 1,564), HFPC (n = 5,112), EC (n = 870), OTHER (n = 4,804), and absences (Abs; n = 1979). The class OTHER included three main categories: broadband calls which cannot be clearly identified as any of the previous classes, whistles and overlapping calls which cannot be distinguished. The class Abs included snippets [of length randomly (uniformly) chosen between 0.4 and 2.5 s] with no vocal activity. Each sample (not labeled as Abs) contained one single labeled call. The second, or whistle dataset, was built from samples from the call dataset but exclusively comprising information about the presence (n = 3587) and absence (n = 1990) of whistles, for a total of 5577 snippets. Finally, a third or test dataset was made from three 10-min audio files fully labeled unseen in the call and whistle datasets, in order to evaluate the performance of detection and of the complete workflow on new data. These files were chosen to each represent a different scenario: low, medium, and high call density (Table I).

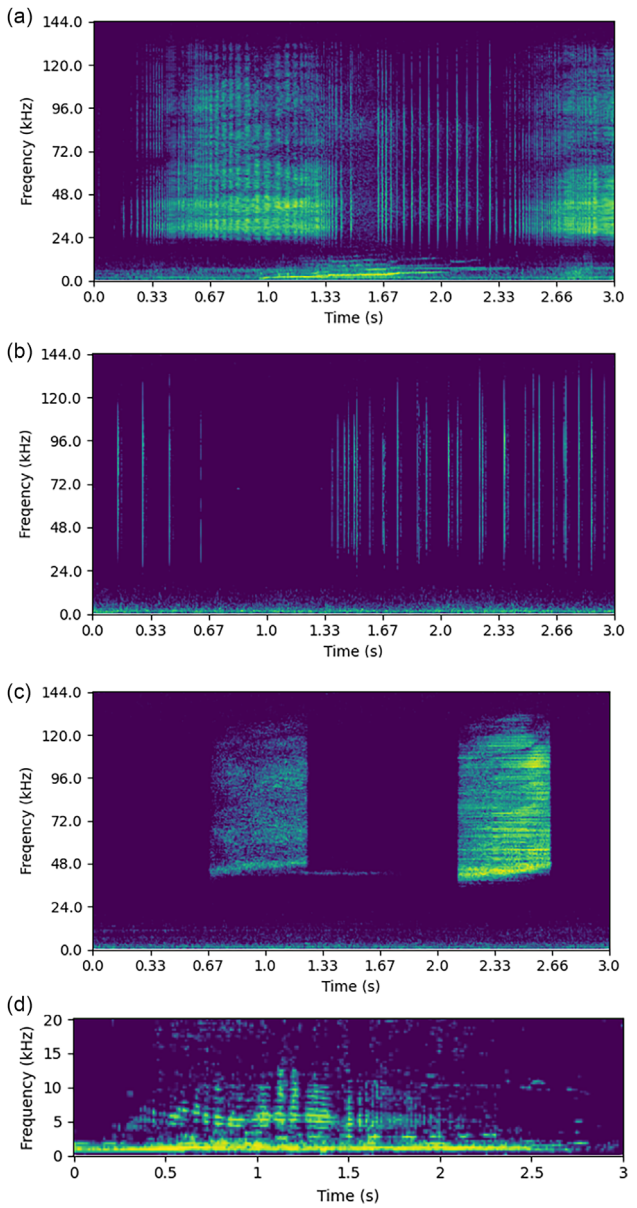

FIG. 1. (Color online) Spectrograms showing the SLEB's call types classified in this study: (a) CC, (b) EC, (c) HFPC, and (d) whistle. The CC, EC, and HFPC spectrograms correspond to a 3-s snippet with 0–144 kHz frequency range. The whistle spectrogram focuses on 0–20 kHz frequency range. All spectrograms show preprocessed acoustic recordings.

B. Audio preprocessing

The presented detection or classification algorithms takes as input audios (waveforms) or images (see Fig. 1).

Audio recordings used directly as input were not preprocessed, except for an optional decimation. On the other hand, preprocessed spectrograms were used as input images. There was a general common preprocessing scheme applied to all spectrogram images (see Fig. 2).

Amplification was used to improve the SNR between the vocalizations and the background noise, and consisted of applying the following function a to each black and white pixel in the image: $a(x) = \log(1 + cx)$, where x is the intensity of a pixel and c is an amplification constant parameter.

When converting to decibel, a minimum threshold, fixed to a minimum decibel level, was applied to remove background noise. We used a min-max scale normalization (normalize values between 0 and 1 by removing the minimum value and dividing by the range of taken values). Details of the other pre-processing steps and the parameters used at each step (according to the corresponding modeling algorithm) are given in Secs. II C and II D. Examples of computed spectrograms following these preprocessing are shown in Fig. 1. The details of the used parameters for each case are presented in Appendix A.

TABLE I. Sample distribution per call type and audio file-testing dataset.

Call type	CC	EC	HFPC	Total
Low	0	35	39	74
Medium	8	64	110	182
High	85	39	296	410

C. Detection

1. Models

a. ROI. ROI detection is based on a simple algorithm that searches for regions with the highest acoustic energy in a spectrogram image. To determine the ROIs of a given spectrogram, the image is first blurred and then masked by a double thresholding method. The blurred images are obtained by convolving a Gaussian kernel (with standard deviation std and size 9×9) with a matrix formed by the intensity values of the pixels in the spectrogram images. Double thresholding involves first applying a signal intensity threshold thigh (ranging from 0 to 1) to select pixels with high intensity value. These selected pixels are called seeds. From these seeds, pixels connected to them with value higher than a second threshold tlow (ranging from 0 to 1) are aggregated. These new pixels also become seeds and the aggregating process continues until no more new pixels are aggregated, meaning that there are no more connected pixels with value upper than the second threshold value tlow. In addition, a minimum area min_{roi} is fixed for each ROI. min_{roi} value was always fixed to half the average size of a single echolocation click (40–120 kHz in frequency and 0.05 s in time). The ROI algorithm is not directly trained but its parameters have to be chosen correctly to limit background detection. The ROI algorithm was tested with a few parameter combinations from high to low sensitivity (denoted as high-ROI, medium-ROI, and low-ROI; Appendix B, Table XI). high-ROI parameters were chosen to be very sensitive and maximize detected calls while low-ROI ones were adjusted to at least detect calls with maximum amplitude.

The ROI algorithm was applied to preprocessed images, including a 20–90 kHz bandpass filter and a relatively low amplification value, in order to avoid exacerbating background noise, especially for empty audio recordings (Appendix A, Table X). Low frequency vocalizations (<20 kHz) were consequently not detected with this method.

b. Detection transformer. DETR is an object detection model based on a transformer architecture, capable of detecting and classifying specific objects in an image [for example, a dog, a cat, or an apple (Carion *et al.*, 2020)]. More importantly, DETR is capable of dealing with calls overlapping in time or frequency (as detecting overlapping objects in an image was included in its initial training). DETR was preferred over other models with CNN-only architectures [for instance, YOLO (Redmon *et al.*, 2016)] because it includes a CNN encoder and state-of-the-art attention layers (Vaswani *et al.*, 2023) that boost detection performance (Carion *et al.*, 2020).

We trained and used DETR as a detector, that is, only to trace a bounding box around signals of interest (i.e., any SLEB call). Train and validation data (70%/30% split) consisted of 3-s fully labeled spectrogram images.

We fine-tuned DETR on 8000 3-s spectrogram images labeled with all calls from the call dataset. The start of each 3-s snippet was uniformly randomized to occur between 0 and 1 s before the beginning of a labeled call. These 3-s snippets could therefore contain complex soundscapes in which more than one call was detected and/or overlapped in frequency and/or time. Snippets from the OTHER category were also included, which means DETR should also be able to detect low frequency signals, such as whistles, contrary to the ROI algorithm (but this has not been evaluated in this work).

Original audios were not downsampled nor bandpassed to keep as many information as possible. This was made possible, contrary to the ROI algorithm, because DETR can distinguish high and low frequencies. The entire frequency range (0–144 kHz) was used in labels and no distinction was made between different call types (Table VIII).

Due to the imbalance in the EC class in the call dataset, we fine-tuned DETR a second time, starting from the results of the first fine-tuning, with 250 additional 3-s spectrogram images (with the same preprocessing) which included 145 EC calls (see the Appendix B 2 for a comparison of performance between these two fine-tuning stages).

Both fine-tunes were done using the code furnished on GitHub (2024). In both cases, batch-size was fixed to 4, backbone (ResNet-50) learning rate to 10^{-5} and other parameters to their default values. The first fine-tuning step was realized in 20 epochs starting from a version of DETR

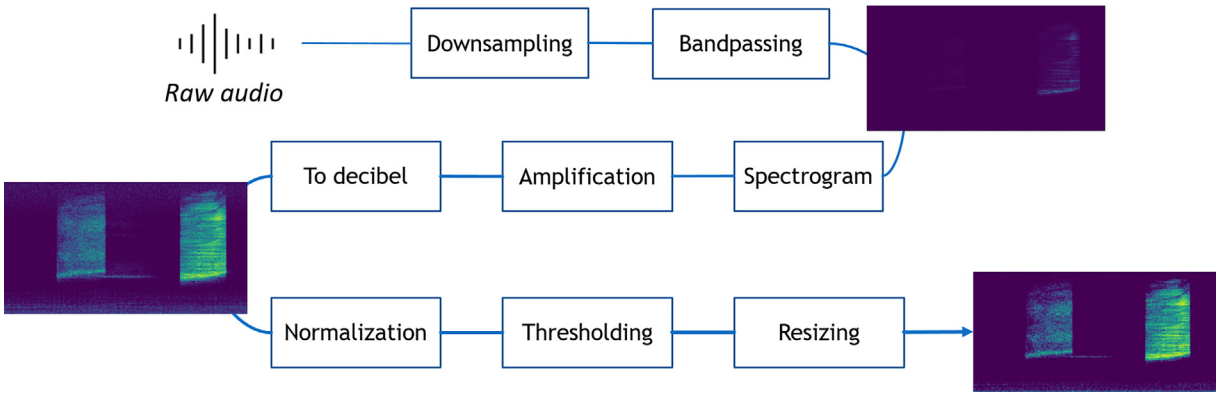

FIG. 2. (Color online) Preprocessing steps from audio to image. Presented spectrograms correspond to a 3-s snippet with 0–144 kHz frequency range. Note that monochrome spectrograms are colorized for better visualisation only.

pre-trained on COCO dataset with no classification head, available online (https://dl.fbaipublicfiles.com/detr/detr-r50-e632da11.pth). A multi-layer perceptron was used as the bounding box output head. The associated learning rate was initialized at 10^{-4} and divided by 10 every 5 epochs. The resulted model (also keeping the output bounding box head) was then trained during the second fine-tuning step for 80 epochs with a head learning rate initialized at 10^{-4} and divided by 10 every 40 epochs. Loss and mean average precision are presented in Appendix B.

A few tests with both bounding box and class predictions were also run (with linear regression as class output head), but gave unsatisfying results, especially a drop in detection performance.

2. Evaluation

The evaluation of the detection algorithms was done on the test dataset. For this purpose, we considered a labeled call to be detected if at least half of its length was among all windows detected.

To estimate how sensitive detection algorithms are at detecting calls, we computed two detection indices: (1) call coverage or *cover*, which is the proportion of the labeled calls length that are detected and (2) call detection d_{all}, which is the proportion of calls detected (i.e., *number* of calls considered detected over the number of all labeled calls). More precisely, the *cover* is defined as follows:

$$cover = \frac{acctime_{det}}{acctime_{all}},$$

where we denote $acctime_{det}$ the accumulated time of all labeled windows considered detected and $acctime_{all}$ the accumulated time of all labeled windows (detected or not). The higher the d_{all} the more calls (in number) have been detected and the higher *cover*, the longer the total duration of detected calls. Cases with high *cover* and low d_{all} could for instance occur when longest calls were almost all detected and many short ones were not. Both these indices take their value between 0 and 1.

Over-prediction occurred when detected windows (also referred to as predicted calls) did not overlap (or the overlap was <50% in length) with the labeled calls. To estimate how much detection algorithms over-predicted, we computed two over-prediction indices: (1) the global overestimation o_{all}, which is defined as the proportion of predicted calls (in time) that did not match any labeled call. These included not labeled calls (incomplete and/or with low SNR) and true absences. (2) the absence overestimation o_{abs}, which is defined as the proportion of predicted calls (in time) that were true absences. More precisely, o_{all} and o_{abs} are defined as follows:

$$o_{all} = \frac{acctime_{\sim det}}{acctime_{all}},$$
$$o_{abs} = \frac{acctime_{abs,\sim det}}{acctime_{all}},$$

where we denote $acctime_{\sim det}$ the accumulated time of all predicted calls matching no labeled call, $acctime_{abs,\sim det}$ the accumulated time of all predicted calls that were true absences, and $acctime_{all}$ the accumulated time of all labeled calls (detected or not). The higher o_{all}, the more unlabeled windows were predicted as a call and the higher o_{abs}, the more windows with absences were detected as calls. These over-predicted calls can be filtered out (classified as absences) in the following classification steps of the workflow. Having o_{all} and o_{abs} with similar values would mean most over-predicted windows were classified absence in the next classification step. It is also interesting to consider $o_{call} = o_{all} - o_{abs}$, which measures the proportion of over-predictions that were additional calls (classified as such) and not labeled. It should be noted that o_{abs}, o_{all}, and o_{call} can be larger than 1. For instance, a o_{all} larger than 2 would mean the total duration of predictions not matching any labeled call exceeded two times the total duration of labeled calls.

D. Call type classification

1. Models

a. CNNs. A residual network structure (ResNet-18) was used as CNN model to classify SLEB vocalizations. The call dataset was used as the training set, but the length of the snippets comprising each labeled call was fixed at 1.5 s, starting at the call onset, to speed up CNN training (see the Appendix C for details on the fixed length). This dataset was split between a train, validation and test with a respective 60%/20%/20% distribution taken in chronological orders (to mimic real training and prediction conditions). The preprocessing used was similar to the one used in DETR but a downsampling and a bandpass filter were applied to the raw audio first (Table IX). Performance modifying these two parameters were also compared and are presented in Appendix C. The final training images were downsampled by 2 and bandpassed with a 0.1–70 kHz filter.

During the training, data augmentation was used (horizontal shift, frequency and time masking) in order to boost model generalization. A random horizontal shift was uniformly sampled between 0 and 0.3 s (threshold chosen so that to have at least half of the considered call 99% of the time). A random frequency masking was randomly located (values set to zero on a continuous range of frequencies) and had a frequency length uniformly sampled between 0 and a quarter of the full bandwidth. Similarly, a time masking was randomly located with length uniformly sampled between 0 and a quarter of the processed call.

The ResNet-18 structure was implemented in PYTHON using PyTorch. It was trained with a weighted cross-entropy loss, which were chosen to balance errors between classes. More precisely, for each class i the associated weight was n_{max}/n_i, where n_{max} was the maximum number of samples over all classes (5112 samples with HFPC class in our case) and n_i the number of samples of class i. We used a weight decay of 0.01, along with an Adam optimizer (parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$). Learning rate was initialized at

10^{-3} and divided by 2 every 3 consecutive epochs without improvement on the validation set. An early stopping criterion about the number of epochs without improvement was also implemented (20 epochs with no improvement on the validation set).

b. AVES. The second single-target classifier tested is based on the AVES algorithm (Hagiwara, 2022). AVES is a model based on a transformer architecture, able to classify calls and identify species vocalizations from raw audios. This model has shown to outperform most of the more common methods such as CNN for call type classification and detection tasks on different species (marine mammals, bats, bird, etc.) (Hagiwara, 2022). It consists in a normalisation layer followed by a CNN encoder and a transformer designed to extract meaningful features from recordings. We can then train a machine learning algorithm on those features to classify the SLEB call types.

The call dataset was used to train and test the algorithm. 75% of the dataset was used for the training and validation of the algorithm while the remaining 25% was used for testing. Training audio samples were obtained by using the exact timestamps for the beginning and the end of each call as they were initially labeled. The preprocessing consisted in a decimation of the audios from 288 kHz to 120 kHz before applying the AVES algorithm in order to limit computation times (see Appendix D).

The fine-tuning was realized using a support vector classifier (SVC) (Schölkopf and Smola, 2001) (see Appendix D for results from other tested classifiers) with a radius based function (RBF) kernel with the following parameter: C = 1 (implementation form scikit-learn).

2. Evaluation

To evaluate and compare single target classification models' performance, we used confusion matrices and BAcc. BAcc is defined as the average of the diagonal values of the associated confusion matrix, and contrary to regular accuracy, is not weighted by the number of samples in each class.

E. Whistle presence classifier

In our workflow we included a parallel classification step to detect the presence/absence of whistles. In this step, instead of working with the detected calls (from detection step) the algorithm directly works with the original recordings preprocessed in a specific way according to the classification model used (CNN or AVES). Since the main energy of SLEB's whistles can be registered between the 0 to 20KHz frequency range, by decimating the original files, whistles can be isolated in the frequency domain from CC, HFPC, and EC.

1. Models

a. CNN. For the same computation reasons as in CNN call type classification, a length was fixed for building the training, validation and testing dataset. This dataset contained all calls in the whistle dataset adjusted to 1.5-s length. Since shorter and longer calls were extended or cut respectively to fit the 1.5-s fixed time window, it occurred that some presence/absence labels had to be adjusted accordingly. The original whistle dataset ended up consisting of 3927 presence and 1650 absence samples. This final dataset was split into train, validation and test with a 60%/20%/20% distribution, respectively.

Input images were preprocessed following the same protocol used for the call type CNN model (Table X). Training was also similar, except that cross-entropy loss weights were chosen balanced and not to favor any class. No data augmentation was used.

b. AVES. AVES model was trained and tested on the whistle dataset with the following partition: 75% of the dataset for the training and 25% for the testing.

Audios were obtained in the exact same way as for AVES call type classification (exact timestamps). We used a SVC (with a RBF kernel) as classification algorithm with parameters $C = 1$ and $max\_iter = 10\,000$ to ensure the convergence of the algorithm.

Computing the AVES features for two different frequencies (one for the call type classification and another for the whistle detection) can be very time consuming and therefore using the same features for both tasks is more efficient. We tested to decimate to two sampling frequencies: 40 kHz, the maximum frequency for low-component calls, and 120 kHz, the frequency yielding the best results for AVES call type classification. Details of the performance of the model with the two decimation schemes and the AVES fine tuning are given in Appendix D).

2. Evaluation

To evaluate and compare the whistle detectors, we used confusion matrices and BAcc.

F. Pipeline

1. Method

The global final workflow consisted in a pipeline comprising the preprocessing, detection, and classification steps in order (Fig. 3). An acoustic file of above 3-s length that enters the pipeline is split into 3-s snippet with a 1.5-s overlapping between them (for instance, for a 1 min file this process would yield $60/3 * 2 = 40$ 3-s snippets). Overlapping between snippets is necessary as splitting a call is very likely to occur when automatically dividing long files in 3 s snippets. Every 3 s snippet is then preprocessed and analyzed to detect and classify SLEB high-frequency call types and whistles. Only calls which begin in the first 1.5 s of a snippet are considered. A call beginning after 1.5 s is expected to be detected in the next snippet. A length of 1.5-s was chosen because this duration exceeds 95% of all labeled call lengths.

https://doi.org/10.1121/10.0030472

We applied several complementary steps to reduce call overestimation. The potential high sensitivity of detection algorithms may split an echolocation event into distinct echolocation pulses. Each echolocation pulse can be brief in duration (down to 0.01 s) and distant from the subsequent pulse within the same echolocation event (up to 0.4 s). Thus, detected ECs that were both short in duration (<0.15 s) and close in time (<0.4 s apart) were merged as one EC. After classification of all detected calls in their respective categories, ECs were again merged if at least one of them lasted less than 0.15 s and if they were separated by less than 0.4 s. This second merge regrouped isolated ECs with longer adjacent echolocation events (since in this second merge, only one of them had to be short enough).

2. Evaluation

a. Detection and classification combined. To evaluate the pipeline we computed similar indices to the one presented in the detection part but applied to specific call types. Similarly, for any call type t we denote $acctime_{det,t}$ the accumulated time of all labeled windows of call type t considered detected and $acctime_{all,t}$ the accumulated time of all labeled windows of call type t (detected or not). Calls coverage for call type t is defined as follows:

$$cover_t = \frac{acctime_{det,t}}{acctime_{all,t}}.$$

We also defined d_t the share of detected calls (in number and not in time contrary to $cover_t$) for any call type t.

To estimate how much detection algorithms overpredict, we computed an over-estimation indice o_t defined as the proportion of calls predicted not detected in labels. Whereas a call was considered detected if at least 50% (in time) was found in labels for detection task, a call was considered well detected (detected and classified correctly) if at least 20% (in time) matched one or multiple labeled calls belonging to the same class. The 50% threshold was chosen empirically before detection results drop. For classification, the threshold was also chosen empirically, but fixed low to detect CC and HFPC when mixed with EC. In the same way as the detection test, these three indices were computed on the test dataset.

b. Computation time. The global pipeline was evaluated in computation time as well. Using DETR or ROI as detector, and AVES or CNN for both call type and whistle classifiers, all combinations were tested. Moreover, two setups were used: with and without GPU NVIDIA GeForce RXT 4090 (with memory size 24GB GDDR6X). GPU computation included all neural networks applications (AVES, DETR, CNN) but excluded spectrogram computation. A CPU Intel(R) Core(TM) i9-9900K (3.60 GHz with 64 Go RAM) was used in both setups. Times were measured on four 45-min audio recordings and averaged. These audio recordings were chosen to correspond to absence, low, medium and high call density, each respectively having 0, 257, 505, and 2378 calls.

III. RESULTS

A. Detection

The detection performance of the ROI algorithm increased with the sensitivity of the parameters chosen (Table II). However, the high-ROI configuration also increased the model's over-detection, reaching a o_{all} of 100% and a o_{abs} of 99% when applied to the low-activity file (Table II).

In both the low and medium activity test audio files, even with the high-ROI configuration, long ECs (longer than 2 s) were partially detected but not enough to be considered detected (Table XVII).

DETR over-detected very few absences in all configurations (Table II). Its performance regarding detection indices (*cover* and d_{all}) increased with the call activity of the audio files and were comparable to the medium-ROI configuration (Table II).

DETR had a bigger gap between *cover* and the percentage of calls detected d_{all} than the medium and high ROI configurations (Table II). It detected some labeled calls in the high activity test audio files that were otherwise missed by ROIs, especially long EC (Table II).

ROI efficiency at detecting independent calls decreased in acoustic contexts with a high proportion of overlapping signals, such as in high call activity test audio recordings. The algorithm did not show a huge increase in number of calls detected between medium and high activity files compared to DETR (Table III). The same applied between medium and high call-density audio files.

Regarding the number of calls predicted, two main phenomena were observed when ROI sensitivity was increased (especially when increasing the standard deviation parameter linked to the blurring step of the ROI algorithm). First, the relative density of predicted calls increased, which caused previously near vocalizations to be merged together, and globally the number of calls decreased (but not d_{all}, which increased). This was seen between the medium-ROI and the low-ROI configurations applied on the low call activity test audio file (Table III). Increasing even more the sensitivity of ROI had the opposite effect, which was seen between the high-ROI and medium-ROI configurations applied on the low call activity test audio file (Table III): low-intensity calls not yet detected started to be predicted which increased the total number of predicted calls.

B. Call type classification

1. CNN

The best ResNet-18 model (with a 1.5-s window and with a 0.1–90 kHz bandpass) had a balanced accuracy of 0.86 (accuracy was 0.85; Table IV).

The CNN showed good predicting performance, particularly in effectively discriminating absence from presence.

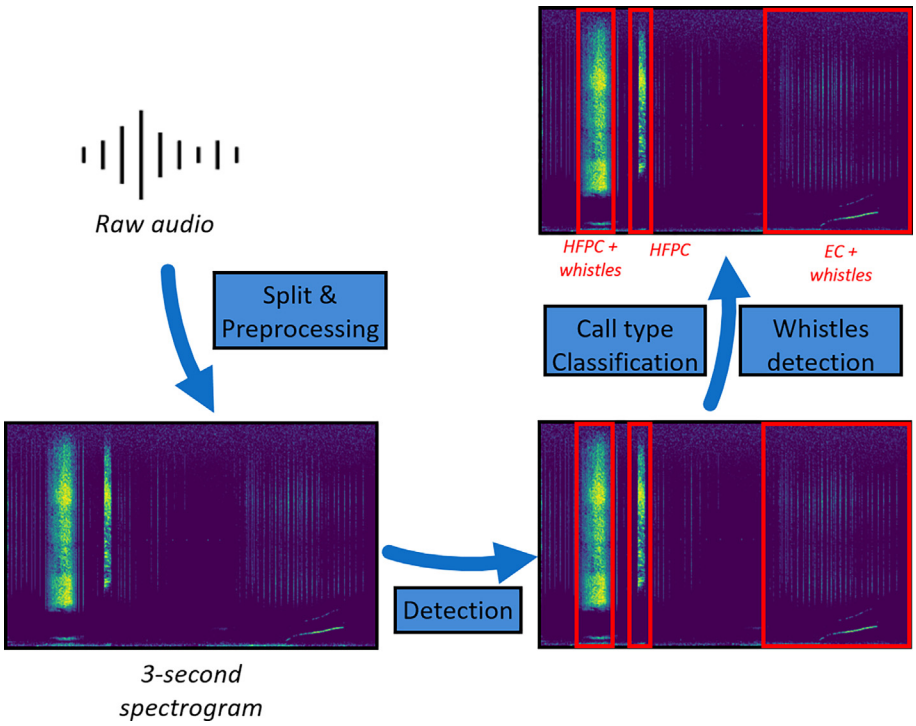

FIG. 3. (Color online) Pipeline steps from input audio to call timestamps, category and whistles presence. When a call was detected (by either DETR or ROI), a call classifier and a whistle classifier were applied. These classifiers had their own preprocessing different from the one used in detection (not represented for clarity). Red bounding boxes correspond to detected calls.

Each call type class showed similar accuracies of around 80%. About two thirds of errors in predicting labeled CC and HFPC calls were due to the model predicting EC instead. This occurred when very dense EC concurred with CC or HFPC in the 1.5-s windows. Similarly, errors in predicting labeled ECs were mostly due to concurrence and predominance of broadband calls in these snippets.

2. AVES

AVES, combined with a support vector classification model, yielded a balanced and global accuracy of 0.87 and 0.93, respectively (Table V). Fine-tuned AVES proved to be very efficient at discriminating absence from presence (0.99 accuracy) as well as CC and HFPC calls. However, 25% of EC calls were predicted as HFPCs (Table V).

C. Whistle detector

1. CNN

The ResNet-18 model achieved a BAcc of 0.82 (and accuracy 0.85) in determining whistle presence/absence from decimated audio files. The unbalanced training dataset favored presence prediction (Table VI).

2. AVES

The performance of the binary classifier in detecting the presence/absence of whistles was high and very similar, whether the AVES features were calculated on decimated acoustic files at 40 or 120 kHz sampling rate. With a decimation of 40 kHz, SVC achieved a BAcc and accuracy of 0.885 and 0.903, respectively (Table VI), and with a decimation of 120 kHz, SVC achieved a BAcc and accuracy of 0.882 and 0.900 respectively (Table VI).

D. Pipeline

1. Performance evaluation of all combinations of detection and classification algorithms

All combinations of detection and classification algorithms achieved accurate results at predicting call types (Table II). Average detection (considering d_{HFPC}, d_{CC}, and d_{EC}) across algorithm pairs was 70% for HFPC and CC calls and 40% for ECs.

Over-estimation increased with call activity in the test dataset and was unbalanced between call type (Table II). For all call types, roughly no over-estimation occurred in the low call activity audio, half the total labeled time was over-estimated in the medium call activity audio, and almost twice as long as the total labeled time was over-estimated in the high call activity audio. CC, which were not abundant in the testing dataset, were systematically overestimated, while HFPCs were only overestimated in high call activity file.

TABLE II. Detection indices, including the percentage of accumulated time detected *cover*, the percentage of detected calls (in number) d_{all}, the global overestimation o_{all}, and the absence overestimation o_{abs} per algorithm and audio test file. All values are in percentage.

Call density	Low				Medium				High			
Indices	*cover*	d_{all}	o_{all}	o_{abs}	*cover*	d_{all}	o_{all}	o_{abs}	*cover*	d_{all}	o_{all}	o_{abs}
DETR	41	56	15	11	49	68	10	6	92	97	43	1
High-ROI	61	63	100	99	76	82	70	63	98	99	54	2
Medium-ROI	43	45	46	46	57	70	59	55	97	99	52	2
Low-ROI	32	32	26	26	35	47	36	36	78	97	33	0

https://doi.org/10.1121/10.0030472

DETR plus AVES achieved the best prediction and over-estimation results (Table II). ROI plus CNN had slightly lower detection results with HFPC and CC but much worse with EC, even though AVES was less precise than CNN for this specific category.

2. Computation time

The computation time for the complete pipeline using a CPU was multiplied by 5 when DETR was used compared to ROI (Tables XV and XVI). DETR plus CNN took slightly more than 1 h on the high call activity audio (a 45-min audio file) whereas ROI plus CNN took 5 min (Table XVI). On average, DETR plus AVES largely exceeded real-time capacities (which was expected due to the very high number of weights in both these models). With a GPU, all calculation times were around 10 times faster than the length of the original audio file (45 min).

IV. DISCUSSION

A. Detection

The performance of both ROI and DETR in detecting SLEB calls was high. Compared to DETR, ROI showed higher sensitivity and systematically overestimated the presence of beluga calls. However, higher sensitivity is positive for ensuring the detection of animals in the wild. In our case, higher sensitivity was preferred in this first stage of the overall workflow, as the classification algorithms could perform a second filter later and discard absent or unclear calls. Over-estimation decreased when applying DETR. This result could be expected since the structure of the algorithm allowed it to directly learn to detect specific patterns in the spectrogram images.

Both DETR and the highly sensitive ROI configuration made predictions close to nearly every labeled call in the test dataset. Calls labeled and not considered detected in our evaluation were almost all long calls which lacked the required 50% overlap with predicted calls. Both algorithms performed well to predict unclear cases with multiple overlapping calls. DETR mainly filtered out very short calls but detected all long ones (≥ 0.3 s), even if a clear call type could not be identified. This result suggests that DETR could be used to detect signals of interest that it has not been trained to detect, i.e., that it has never seen before, such as SLEB call types that we haven't considered in this study or even vocalizations from other marine mammal species (also not considered here). ROI will be sensitive to these extraneous calls (as it relies on spectrogram activity). As for DETR, it could either be trained to avoid or to detect and recognize additional call types or acoustic signals from other sources, providing enough acoustic data.

Regarding ROI, using a highly sensitive configuration with an accurate presence-absence call type classifier seems to be the best option, as some calls with low amplitude could be missed otherwise. However, the different parameters should be adapted to each new acoustic dataset, as background noise and hydrophone setup are expected to change.

The first fine-tuning of DETR using the call dataset probably resulted in a better initialization for its head and backbone and thus in the identification of more optimal parameters that could better characterize the beluga vocal repertoire. However, the effect of fine-tuning twice rather than once has not been studied in depth and, consequently, the role of the first fine-tuning on the model's performance must be interpreted with caution.

Training DETR for the classification task (rather than just using it for detection) was not successful so far in our case. The relatively low number of samples and the presence of the OTHER class made its training complex if not impossible for now. Yet, given the performance DETR had in its original paper (Carion *et al.*, 2020), it should be possible to use it for detection and classification of beluga vocalizations, provided that enough data is available for training.

ECs were generally detected less accurately than CC and HFPC in our dataset. The fact that CC and HFPC were given priority over EC when they coincided in the same labeled window, could explain this result, since the model could have learned to "ignore" single echolocation pulses. Indeed, complex acoustic compositions, including overlapping calls, were quite frequent in the dataset (reflecting BSM's belugas high social and vocal activity) and only dense and independent EC events were labeled as ECs. When echolocation pulses overlapped with CC and HFPC, they were not considered in the label.

The ROI algorithm has the advantage of being adaptable to a more or less sensitive labeling scheme by modifying its configuration. On the other hand, DETR is expected to predict according to the threshold it was trained with, and would need to be re-trained and tested for a more or less sensitive version. In all cases, the potential gap existing between the detector threshold and the one used for manually labeling in this study could explain why there is some overestimation in the detection results, especially for audio

TABLE III. Number of calls labeled and predicted for each detection algorithm on the audio test files with three different call density levels.

Call density	Low	Medium	High
Label	74	182	410
DETR	155	348	1100
High-ROI	550	560	431
Medium-ROI	230	511	470
Low-ROI	421	645	546

TABLE IV. Confusion matrix—Call type classification with CNN.

		Prediction			
Testing set		Abs	CC	EC	HFPC
Label	Abs	0.998	0.000	0.000	0.002
	CC	0.027	0.821	0.101	0.051
	EC	0.029	0.081	0.814	0.076
	HFPC	0.023	0.065	0.105	0.807

TABLE V. Confusion matrix—Call type classification with AVES.

Testing set		Prediction			
		Abs	CC	EC	HFPC
Label	Abs	1.000	0.000	0.000	0.000
	CC	0.018	0.888	0.026	0.068
	EC	0.005	0.108	0.637	0.250
	HFPC	0.002	0.014	0.017	0.967

recordings with a high call density. In these cases, the sensitivity of the manual labeling may have diminished and caused an increase in the pipeline's overestimation rate.

Developing an automated tool to detect and classify SLEB acoustic activity reduces the subjectivity of manual procedures and the discrepancy that could exist in the detection/classification between human observers. Nevertheless, supervised machine learning methods, as the ones developed in this study, need to be trained with datasets that are manually analyzed. They are therefore still subject to relative subjectivity regarding the threshold that is used to define and label the calls in the "ground truth" dataset. In addition, manual procedures remain more accurate than automatic methods for now, especially when many calls overlap.

B. Classification

The two classification models (AVES and CNN) were very efficient at automatically distinguishing between the presence and absence of high-frequency call types in BSM recordings (accuracy greater than 99%).

AVES showed a better BAcc than CNN when distinguishing between call types, but its performance was less even across classes. It was more prone to predict HFPC to the detriment of EC. AVES performed also slightly better for the whistle detection in classifying calls short in time. Both classifiers achieved optimal performances when trained and tested on decimated data in the range of 150–200 kHz sampling frequency. BAcc only varied 1% when decimating to 120 kHz with AVES. This suggests that a lower sampling rate (<288 kHz) could be used with this model to detect the presence and classify the acoustic activity of belugas at BSM. A lower sampling rate would have the added benefit of extending the life of the passive acoustic monitoring devices. We hypothesize from these results that the signature of a high frequency call can be detected at their rather low frequency parts.

In the pipeline, the detection algorithms were more sensitive than the classification ones. This allowed an increase of the detection capabilities of the workflow and narrow down the classification to the SLEB call types of interest (labeled calls). However, this also implies that on new data, the classifiers will not be able to identify new calls (not labeled in the current dataset and not included in the training phase). Models may either classify them as absences or misclassify them in another category. In order to classify new call types, the call classifier (AVES or CNN) would need to be retrained.

In this study, we sought to detect and classify four main call classes. Some of these classes encompass a wide variety of vocalizations (e.g., complex contact calls and OTHER calls). Further work could add more flexibility to the current pipeline, for instance by training classification models to distinguish between calls within the same broad category or by linking high-frequency calls to low-frequency components. This could allow us to discriminate between calls such as simple and complex CC. Being able to automatically and specifically identify complex CC from long-term or real-time recordings, could provide very valuable information to better understand SLEB herd structure and group composition (Vergara *et al.*, 2021).

TABLE VI. Confusion matrix—Whistle detection with AVES and decimation to 120 kHz.

Model	Acc	BAcc	Absence acc	Presence acc
CNN	0.850	0.820	0.749	0.897
AVES—40 kHz	0.903	0.885	0.819	0.951
AVES—120 kHz	0.900	0.882	0.816	0.948

C. Pipeline

The versatility of the pipeline proposed in this study lies in the different algorithms that can be used at each stage. Whether the aim is to modify the sensitivity of the detection phase, increase accuracy to detect a specific call type, or optimize calculation time, users can adapt the workflow to their needs.

ROIs and CNN are fast to compute but less accurate generally speaking than DETR and AVES, respectively. Indeed, even though DETR multiplies computation time by 5, it is capable of detecting single whistles and managing overlapping calls. This computation time issue can be worked around with a GPU for offline analysis. However, using a GPU for real-time computation (i.e., real-time) is costly and more complex to install and manage on-site. For CPU-only real-time computation, the ROI-CNN pipeline represents the most convenient choice.

Indices computed to evaluate the entire pipeline should be taken carefully. We used data that models were never

TABLE VII. Preprocessing parameters for ROI spectrogram image computation.

Step	Parameter	Value
Downsampling	coefficient	2
Bandpassing	frequency range	20–180 kHz
Spect	window name	Hamming
Spect	window size	5ms
Spect	overlap	25%
Amplification	c	10
To db	minimum db level	−30
Normalisation	method	minmax
Thresholding	method	0.3
Resizing	output size	(512, 256)

TABLE VIII. Preprocessing parameters for DETR spectrogram image computation.

Step	Parameter	Value
Downsampling	coefficient	X
Bandpassing	frequency range	X
Spect	window name	Hamming
Spect	window size	5ms
Spect	overlap	25%
Amplification	c	10^7
To db	minimum db level	0
Normalisation	method	minmax
Thresholding	method	0.8
Resizing	output size	(512, 256)

TABLE X. Preprocessing parameters for CNN whistle spectrogram image computation.

Step	Parameter	Value
Downsampling	coefficient	6
Bandpassing	frequency range	0.5-40 kHz
Spect	window name	Hamming
Spect	window size	5ms
Spect	overlap	25%
Amplification	c	10
To db	minimum db level	−30
Normalisation	method	minmax
Thresholding	method	0.3
Resizing	output size	(256, 64)

trained on to construct the test sets to assess the performance of each fully trained model. While this increased the robustness of the model's performance results, it also led to a final test set (to assess the entire workflow performance) with fewer samples. Nevertheless, general results followed expectations (DETR plus AVES combination had the best results) and should not change with more similar test data.

While the workflow has shown its robustness with acoustic data from BSM it will be necessary to test the pipeline performance on new data from other regions of the SLEB habitat to generalize our results to all high residence areas in the St. Lawrence Estuary (SLE). Indeed, this tool would be most useful to continuously and automatically monitor SLEB activity, at locations where SLEB presence and acoustic behavior are strongly related.

To our knowledge, there are no examples in the literature of a complete pipeline capable of automatically detecting and classifying the acoustic activity of belugas, covering the entire frequency range of its acoustic repertoire and continuously providing the duration (start and end) and number of calls. Few automatic call detection tools have been developed so far to analyze acoustic recordings from beluga whale vocalizations. Those that exist are focused on detecting low-frequency components of their repertoire and on a fixed time-window (Erbe and King, 2008; Zhong *et al.*, 2020). In general, deep learning structures have shown to outperform other detection and classification tools used so far to detect the presence/absence of acoustic signals of interest from PAM data. Moreover, most detection algorithms are still trained to detect presence on a fixed time-window (Bergler *et al.*, 2022; Schröter *et al.*, 2019). Several studies have shortened the temporal windows over which the detection occurs and aggregate results later on (Roch *et al.*, 2021). Nevertheless, these methods are normally species-specific and can rarely be generalized to other vocal repertoires. Applying state-of-the-art detection algorithms such as transformers (e.g., DETR) to accurately detecting calls from spectrogram images has not been tried yet in bioacoustics. This approach has great potential and could be tested with other vocal species or any sound detection problem requiring accurate (in time) predictions with varying length. Deep learning and in particular CNN are also increasingly being used to automate the classification of animal vocalizations, including marine mammals (Bergler *et al.*, 2019; Bermant *et al.*, 2019). For beluga calls, some machine learning models have been used to classify acoustic presence [(Booy *et al.*, 2021); overall accuracy 85%] or validate manually grouped call types [(Garland *et al.*, 2015); overall accuracy 83%]. However, they have only focused on the low-frequency components of the repertoire, and the generalization and applicability of these models have yet to be tested. Our classification model outperforms these models in terms of accuracy (87%) and the call types it is able to discriminate automatically (four, including low frequency signals), which also cover the frequency range of SLEB vocalizations. On the other hand, it is worth noting, in terms of future applicability, that high frequency signals are absorbed more rapidly so the range at which our model could detect HPFC or EC would be considerably reduced compared to low frequency signal detectors.

TABLE IX. Preprocessing parameters for CNN call type spectrogram image computation.

Step	Parameter	Value
Downsampling	coefficient	2
Bandpassing	frequency range	0.1–180 kHz
Spect	window name	Hamming
Spect	window size	5ms
Spect	overlap	25%
Amplification	c	10^7
To db	minimum db level	0
Normalisation	method	minmax
Thresholding	method	0.8
Resizing	output size	(512, 256)

TABLE XI. Chosen parameters for different ROI configurations. Gaussian blur was applied with standard deviation *std*, and the double thresholding with the high threshold t_{high} and low threshold t_{low}.

Configuration	High-ROI	Medium-ROI	Low-ROI
std	5	2	5
t_{high}	0.01	0.01	0.3
t_{low}	0.005	0.005	0.2

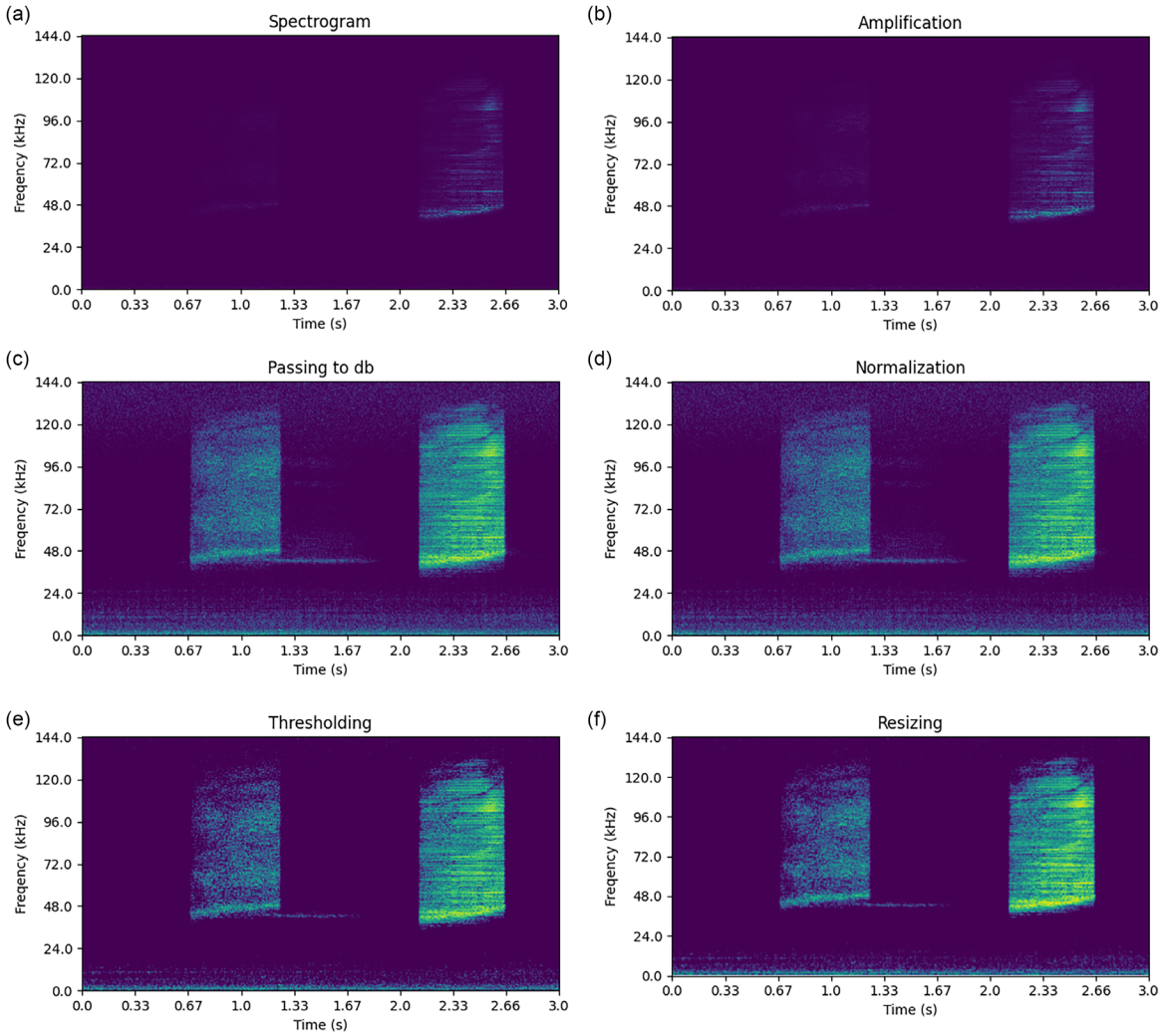

FIG. 4. (Color online) Preprocessing steps of an audio with two HFPCs computed with parameters detailed in Table VIII.

TABLE XII. Detection indices, including the percentage of accumulated time detected *cover*, the percentage of detected calls (in number) d_{all}, the global overestimation o_{all}, and the absence overestimation o_{abs} per DETR training method and audio test file. All values are in percentage.

Call density	Low				Medium				High			
Indices	o_{all}	o_{abs}	*cover*	d_{all}	o_{all}	o_{abs}	*cover*	d_{all}	o_{all}	o_{abs}	*cover*	d_{all}
DETR	15	11	41	56	10	6	49	68	43	1	92	97
DETR only first fine-tune	0.8	0.5	4	21	0.4	0	8	29	8	0.2	35	54
DETR only second fine-tune	11	7	46	46	7	4	45	65	44	1	90	97

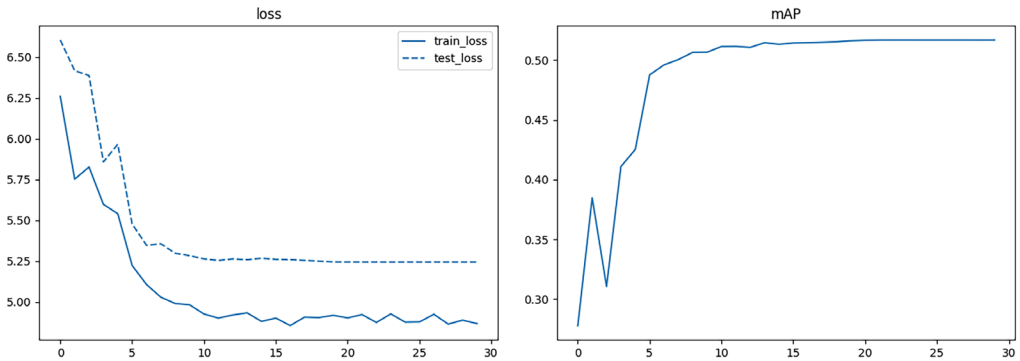

FIG. 5. (Color online) DETR loss (left) and mean average precision on validation (right) per epoch of the first training with the partial dataset.

https://doi.org/10.1121/10.0030472

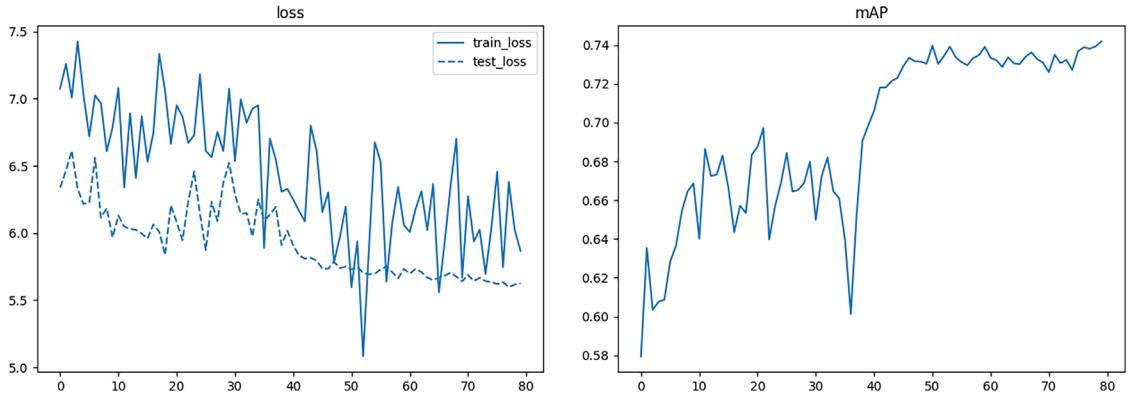

FIG. 6. (Color online) DETR loss (left) and mean average precision on validation (right) per epoch of the second training with the complete dataset.

Interesting machine learning environments have also emerged, offering publicly accessible, species-independent tools that can be trained and used to detect specific acoustic signals, discriminate between acoustically active vocal species in recordings or recognize call types (Bergler *et al.*, 2022). However, few studies have addressed the growing need to develop complete workflows that automatically perform signal detection and classification tasks on a continuous basis. For beluga whales, to our knowledge, only one other study has developed such a tool, achieving an overall accuracy of 98% and 88% for detection and classification, respectively (Miralles *et al.*, 2013), which is very similar to our results (97% and 87%, respectively). However, Miralles *et al.* (2013) trained and tested their models with recordings from only two individuals, recorded under the controlled conditions of an aquarium, whereas we have already tested our pipeline under natural conditions, using two years of recordings. Complete and robust workflows, such as the one proposed in this study for SLEB, are needed to make machine learning tools useful in conservation practice, to be included in real-time monitoring programs and to inform dynamic conservation plans.

The pipeline we propose here could have many different applications. Long term PAM data could be rapidly analyzed to provide useful information on the spatiotemporal distribution and habitat use patterns of SLEB based on vocal activity in the SLE. It is true, however, that like any method based on acoustic activity, the utility of the pipeline in a particular environment would be limited by the consistency of the SLEB's vocalizations over time and space. Although our understanding of the functionality of the SLEB call types is still at an early stage, we already know that the four main categories of calls that we have trained the pipeline to discriminate are recurrent and some (echolocation clicks, for instance) are good indicators of relevant behavioral states of belugas. Low-frequency modulated and pulsed whistles or tones are frequently produced during social interactions and directional swimming (Belikov and Bel'kovich, 2003; Garland *et al.*, 2015; Panova *et al.*, 2012), and although their specific function is not yet well known, their detection also extends the spatial range over which SLEB can be detected, since these low-frequency signals are the ones that travel further in the water (Vergara *et al.*, 2021). CC are used for group cohesion and between mother and calf pairs to keep in contact (Vergara *et al.*, 2010). Complex CC may even allow one to discriminate individual vocal signatures (Vergara and Mikus, 2018). Training the current workflow to identify and count complex CC could therefore be a very relevant step towards automatically estimating the composition of SLEB groups, or even their local abundance. This pipeline is therefore a valuable tool that could represent a step forward in the investigation of the beluga call functioning, as continuous, standardized and real-time labeled records of their vocal activity is a key element in reconstructing the

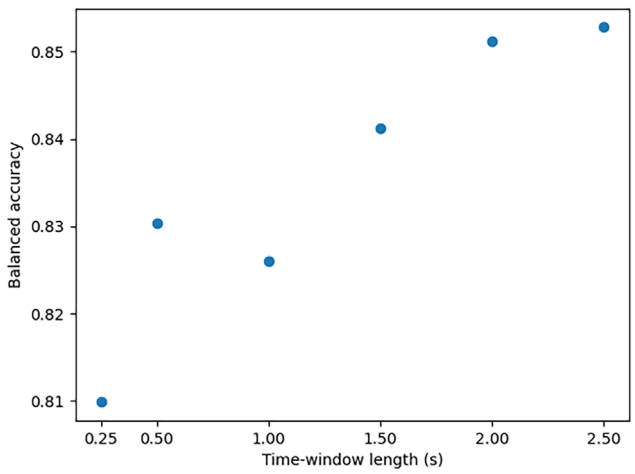

FIG. 7. (Color online) Balanced accuracy according to selected fixed window length (in second) for CNN call type classification.

TABLE XIII. Obtained balanced accuracies with different bandpass frequency range applied in preprocessing.

Frequency min (kHz)	Frequency max (kHz)	Balanced accuracy
20	90	0.841
20	70	0.838
20	50	0.821
0.1	90	0.860
0.1	70	0.848
0.1	50	0.853
0	144	0.858

TABLE XIV. Details of all tested machine learning algorithms along with their parameters for AVES grid search.

Algorithms	Parameters range
K-nearest neighbors	N-neighbors: 2–10
Random forest	N-trees: 20–100 max depth: 2–8
Ridge classifier	alpha: $10^{-2} - 10^{1}$
Ada boost classifier	N-trees: 40–100 learning rate: $10^{-2} - 10^{0}$ trees' depth: 1–5
Gradient boosting classifier	N-estimators: 40–90 learning rate: $10^{-1} - 10^{0}$ trees' depth: 2–10
Linear SVC	C: $10^{-3} - 10^{0}$
Polynomial SVC	C: $10^{-1} - 10^{1}$ degree: 2–3
RBF SVC	C: $10^{-2} - 10^{0}$
Sigmoid SVC	C: $10^{-2} - 10^{0}$ Eta: $10^{-1} - 10^{0}$
Tree based XGB classifier	Alpha: $10^{-1} - 10^{0}$ Lambda: $10^{-1} - 10^{0}$ trees' depth: 2–5
Linear based XGB classifier	Alpha: $10^{-3} - 10^{0}$ Lambda: $10^{-3} - 10^{0}$

TABLE XV. Pipeline mean computation times of different detection and classification combinations with CPU and GPU.

Device	Detection	Call type classifier	Whistle detector	Computation time (s)
CPU	DETR	AVES	AVES	3337
CPU	DETR	CNN	CNN	2089
CPU	ROI	AVES	AVES	883
CPU	ROI	CNN	CNN	372
GPU	DETR	AVES	AVES	226
GPU	DETR	CNN	CNN	222
GPU	ROI	AVES	AVES	222
GPU	ROI	CNN	CNN	218

link with beluga behavior, in addition to detailed behavioral observations.

V. CONCLUSION

The automatic pipeline that we present in this study constitutes, to our knowledge, the first capable of detecting and classifying SLEB call types from long term acoustic recordings. The pipeline has proven to be very powerful and provides a rapid tool to standardize call type discrimination. It could also be implemented to continuously analyze underwater SLEB recordings in real time. The information that could be directly collected with this pipeline and the derived outputs could greatly enhance our comprehension of the SLEB spatiotemporal distribution and habitat use patterns and, eventually, behavior (when we better understand the relationship between vocal categories and behavior in this species). Furthermore, it could allow one to assess and quantify the functional relationship between SLEB presence and vocal activity and ship transit. Effective real-time monitoring tools offering rapid access to information on the presence and/or behavior of belugas (including vocal activity) could be very useful in informing dynamic measures aimed at mitigating the impacts of shipping traffic in the St. Lawrence Estuary, that could be both effective for the recovery of the SLEB and realistic for industry, such as, slow-down measures or shifting shipping lanes.

ACKNOWLEDGMENTS

Tristan Cotillard and Xavier Sécheresse contributed equally to this work. The authors would like to acknowledge Réseau Québec Maritime (RQM) for funding and support during this study (Grant No. PLAINE-2022PS08).

AUTHOR DECLARATIONS

Conflict of interest

The authors declare no conflict of interest for any aspect of this study.

DATA AVAILABILITY

Data is available from the authors upon reasonable request and with the permission of GREMM and Raincoast Conservation Foundation.

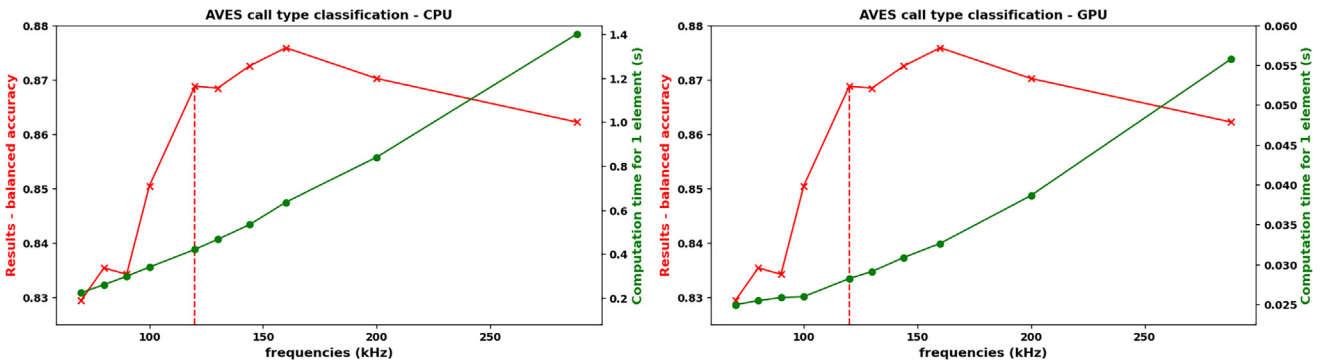

FIG. 8. (Color online) Balanced accuracy (red curve) and computation time (green curve) with CPU-only (left) and GPU (right) configurations for AVES call type classification per frequency used for decimating input audios (initially recorded at 244 kHz).

https://doi.org/10.1121/10.0030472

TABLE XVI. Pipeline all computation times of all algorithm combinations on all audio files with and without GPU.

Combination	Detection	Call type classifier	Whistles presence	Call density	Time (s) GPU	Time (s) CPU
1	DETR	AVES	AVES	Abs	175	2320
1	DETR	AVES	AVES	Low	157	1896
1	DETR	AVES	AVES	Med	237	3740
1	DETR	AVES	AVES	High	334	5392
2	DETR	AVES	CNN	Abs	246	2489
2	DETR	AVES	CNN	Low	215	2055
2	DETR	AVES	CNN	Med	322	3797
2	DETR	AVES	CNN	High	484	5813
3	DETR	CNN	AVES	Abs	246	2483
3	DETR	CNN	AVES	Low	221	2127
3	DETR	CNN	AVES	Med	319	3987
3	DETR	CNN	AVES	High	465	6198
4	DETR	CNN	CNN	Abs	201	1719
4	DETR	CNN	CNN	Low	155	1067
4	DETR	CNN	CNN	Med	223	1997
4	DETR	CNN	CNN	High	307	3571
5	ROI	AVES	AVES	Abs	233	847
5	ROI	AVES	AVES	Low	139	348
5	ROI	AVES	AVES	Med	205	715
5	ROI	AVES	AVES	High	309	1620
6	ROI	AVES	CNN	Abs	305	915
6	ROI	AVES	CNN	Low	197	411
6	ROI	AVES	CNN	Med	281	820
6	ROI	AVES	CNN	High	428	1812
7	ROI	CNN	AVES	Abs	285	1267
7	ROI	CNN	AVES	Low	195	1270
7	ROI	CNN	AVES	Med	276	1756
7	ROI	CNN	AVES	High	414	2749
8	ROI	CNN	CNN	Abs	214	383
8	ROI	CNN	CNN	Low	150	263
8	ROI	CNN	CNN	Med	212	339
8	ROI	CNN	CNN	High	293	501

APPENDIX A

1. Detailed parameters for each preprocessing

All input audios had a sampling rate of 288 kHz. The detailed parameters for every preprocessing is detailed in Tables VII–X.

2. Preprocessing spectrogram examples

The following images represent all important steps of the preprocessing (excluding downsampling and band-passing) computed with the parameters detailed in Table VIII and Fig. 4. Two HFPCs can be seen in this example.

APPENDIX B

1. ROI: Details of the configuration parameters

See Table XI.

2. DETR complementary results

Table XII shows detection performance with one or two fine-tuning steps.

The low number of labeled EC in the call dataset gave unconvincing results, especially the coverage, with only the first fine-tuning step. On the contrary, doing only the second fine-tune gave interesting results, close to DETR twice fine-tuned (see Figs. 5 and 6). Still, this final version had better coverage and percentage of calls globally and a higher sensitivity (which comes with more over-detection).

APPENDIX C

1. Sensitivity to the length of the training snippets

The CNNs' performance increased with the length of the snippets comprising each labeled call type in the training dataset (Fig. 7). 1.5-s window is satisfying trade-off between

TABLE XVII. Detection indices per call type t, including the percentage of accumulated time detected $cover_t$, the percentage of detected calls (in number) d_t, the global overestimation o_t per algorithm and audio test file. The same algorithm has been used for call type classification and whistle detection has been used, and so the combination names have been shortened.

		HFPC			CC			EC		
Combination	File	$cover_{HFPC}$	d_{HFPC}	o_{HFPC}	$cover_{CC}$	d_{CC}	o_{CC}	$cover_{EC}$	d_{EC}	o_{EC}
ROI-CNN	Low	49	50	6.6	0	0	2 s[a]	37	38	2
ROI-AVES	Low	45	55	7	0	0	1.1 s[a]	42	44	1
DETR-CNN	Low	41	47	4	0	0	10 s[a]	38	31	8
DETR-AVES	Low	48	55	10	0	0	3 s[a]	31	18	0.4
ROI-CNN	Medium	67	68	54	61	57	100	24	32	2
ROI-AVES	Medium	61	69	24	66	86	102	42	65	7
DETR-CNN	Medium	58	63	29	55	57	97	26	40	7
DETR-AVES	Medium	62	68	8	59	71	74	83	42	9
ROI-CNN	High	85	86	220	78	81	190	53	56	79
ROI-AVES	High	61	59	145	93	95	208	57	64	68
DETR-CNN	High	79	81	149	73	76	158	57	67	56
DETR-AVES	High	88	93	146	80	85	120	72	85	86

[a]All values are in percentage except these as there is no CC in the low call-density audio file—these numbers correspond to the total time of CC detected.

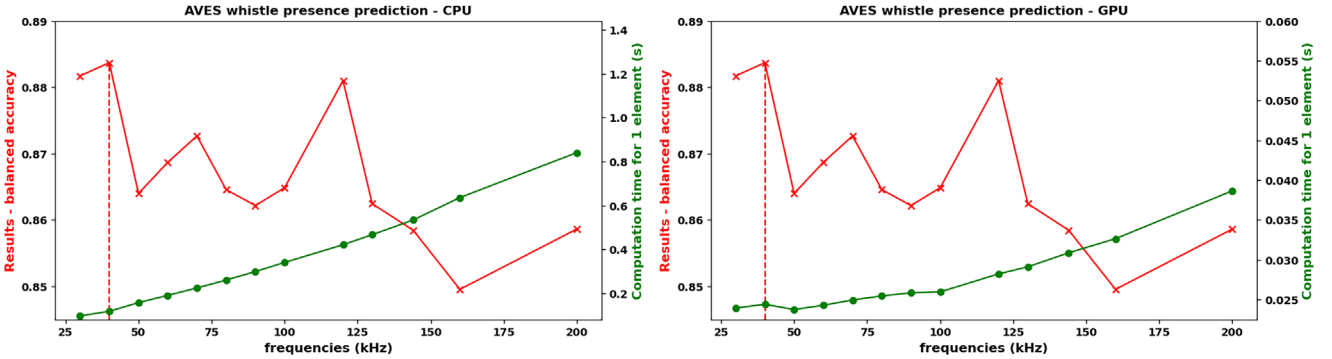

FIG. 9. (Color online) Balanced accuracy (red curve) and computation time (green curve) with CPU-only (left) and GPU (right) configurations for AVES whistle classification per frequency used for decimating input audios (initially recorded at 244 kHz).

performance and granularity (the longer the window the less granularity the pipeline has). 2-s and 2.5-s window had unbalanced confusion matrices, with a gain in performance with EC and a light drop with HFPC and CC.

2. Sensitivity to the bandpass scheme

The CNNs classification accuracy varied with the frequency range of the bandpass filter applied in the preprocessing step (Table IX). A downsampling coefficient of 2 was applied to every preprocessing when possible (with maximum frequency range less than 288 kHz/4 = 72 kHz). Window length was fixed to 1.5 s. Results show that higher accuracy scores are obtained when training with lower frequency ranges and thus for classification, sampling rate used to collect data could be diminished (see Table XIII).

APPENDIX D

1. How to find an optimal classifier?

The fine-tuning of the AVES algorithm was done using the scikit-learn library in PYTHON. We used a threefold cross-validation grid search in order to find the best algorithm along with its parameters. The metric used in the training of the algorithm is the balanced accuracy due to the imbalanced dataset we used.

The list of the tested algorithms and the range of their parameters is given in Table XIV.

The different possible values for the algorithms were optimized depending on the overfitting of the grid search results (in case of overfitting, we increased the regularisation or we simplified the algorithm like reducing the depth of trees in tree based models). The parameters range were the same for the whistle and the call type classifiers.

2. Optimizing accuracy and computation time

In order to optimize the trade-off between computation time and algorithm performance, raw audios can be decimated (using a low pass filter followed by an interpolation to the desired frequency).

We trained and tested each algorithm using 200 random samples (with an average call length of 0.77 s) for every decimated frequency and extracted the best model. We evaluated model performance by computing the BAcc. The calculations were performed on an intel core i9-10900 CPU (2.80 GHz) and the GPU calculations with an NVIDIA GeForce RTX 3060.

The results for call type classification and whistles detection can be seen in Figs. 8 and 9.

APPENDIX E

See Tables XV–XVII. Computation time is long for absence audio files because amplification followed by normalisation exacerbates background noise and so increase false detections which are then predicted as absence by classification algorithms.

APPENDIX F

1. Features

The first approach we chose for the call type classification is to use ML algorithms (common algorithms like K-nearest neighbors, random forests, etc.) applied on features which were computed on audios and spectrogram. However, the obtained results were not satisfying so we chose instead to use DL techniques.

2. Multi-target classification

In order to reduce the accumulated error in the pipeline and limit the impact of the detection algorithms, we trained another model to create an alternative workflow which does not need to detect precisely the calls.

TABLE XVIII. Sample distribution in the different classes—multi-label data set.

Call type	Abs	CC	EC	HFPC	CC+EC	CC+HFPC	EC+HFPC	CC+EC+HFPC	TOTAL
Nb samples	1979	1073	630	4359	50	1162	155	12	9420

https://doi.org/10.1121/10.0030472

TABLE XIX. Preprocessing parameters for CNN multi-target spectrogram image computation.

Step	Parameter	Value
Downsampling	coefficient	X
Bandpassing	frequency range	5-140 kHz
Spect	window name	hamming
Spect	window size	7ms
Spect	overlap	12.5%
Amplification	c	10^5
To db	minimum db level	0
Normalisation	method	minmax
Thresholding	method	0.7
Resizing	output size	(299, 299)

a. Data set

The multi-label dataset comprised 3-s snippets. The start of each snippet was uniformly randomized to occur between 0 and 1 s before the beginning of a labeled call. These 3-s snippets could therefore comprise complex soundscapes where more than a call type is labeled and/or overlaps in frequency and/or time. The repartition pf the classes is given in Table XVIII.

b. Method

The goal of this algorithm is to assess the presence of each call type in a 3 s snippet.

The preprocessing used to compute the spectrogram is identical compared to the single target classification CNNs. The different parameters are given in Table XIX.

We classified 3 s snippets with a multi target CNN trained on the multi target dataset.

The CNN was implemented in PyTorch and trained with a cross-entropy loss, a weight decay of 0.01, along with an Adam optimizer (parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$). Learning rate was initialized at 10^{-3} and divided by 2 every 3 consecutive epochs without improvement on the validation set. An early stopping criterion about the number of epochs without improvement was also implemented (20 epochs with no improvement on the validation set).

During the training, data augmentation was used (horizontal shift, frequency, and time masking) in order to boost model generalization and a batch size of 32 was chosen to ensure reasonable computational times.

After several trials, we chose to use an Inception$_v$3 CNN structure for this algorithm as it had the best results on the validation set. The use of this CNN imposes the input size of the spectrogram to be (299, 299).

TABLE XX. Confusion matrix—Call type multi-target detection.

		Prediction				Prediction				Prediction	
CC		Abs	PRES	EC		Abs	PRES	HFPC		Abs	PRES
Label	Abs	0.96	0.04	Label	Abs	0.97	0.03	Label	Abs	0.88	0.12
	PRES	0.24	0.76		PRES	0.39	0.61		PRES	0.08	0.92

TABLE XXI. Detailed confusion matrix for every combination—Call type multi-target detection. Each triplet is respectively associated with CC/EC/HFPC presence.

		Prediction							
	Test set	[0,0,0]	[0,0,1]	[0,1,0]	[1,0,0]	[0,1,1]	[1,0,1]	[1,1,0]	[1,1,1]
Label	[0,0,0]	367	19	10	4	3	4	0	0
	[0,0,1]	1	811	29	21	15	51	2	0
	[0,1,0]	0	20	85	7	10	7	1	1
	[1,0,0]	0	7	5	156	1	24	2	0
	[0,1,1]	0	15	6	3	2	3	0	0
	[1,0,1]	0	44	0	27	0	121	0	0
	[1,1,0]	0	0	0	0	0	0	0	0
	[1,1,1]	0	0	0	0	0	0	0	0

c. Evaluation

The metric chosen to compute the efficiency of the algorithms and to train them is the mean balanced accuracy.

Mean balanced accuracy was used as the improvement measure for both these processes (training and testing). It was preferred over exact match ratio due to imbalanced classes in the dataset and the importance of the call detection (invert 2 calls does not count as much as invert one call and one absence). It consists in computing the balanced accuracy for the presence/absence of every call in each snippet and then average it (possibly with some weights if needed to improve the training).

d. Results

From Table XX, we can compute the AvgBAcc and the exact match ratio. The respective values for those metrics are 0.85 and 0.82.

More precisely the confusion matrix in Table XXI shows the different combinations possible and the predictions which were done by the algorithm.

e. Discussion

The results of the multi target algorithm are very promising. The detection accuracy is very high so it can efficiently assess the presence of call in short records. Nevertheless, the algorithm use has several defaults compared to the pipeline presented before: the call types are not precisely detected in time by the algorithm (as we can just assess the presence of a call in 3 s snippets) and, more importantly, the algorithm does not count how many identical calls are in the short snippet (predictions are independent from the number of occurrences of a call type in the snippet). Therefore, the number of predicted calls in a long record can be highly underestimated as we saw in the data that having 3 or more calls in a snippet often happened.

Allen, A. N., Harvey, M., Harrell, L., Jansen, A., Merkens, K. P., Wall, C. C., Cattiau, J., and Oleson, E. M. (**2021**). "A convolutional neural network for automated detection of humpback whale song in a diverse, long-term passive acoustic dataset," Front. Mar. Sci. **8**, 607321.

Au, W. W., Carder, D. A., Penner, R. H., and Scronce, B. L. (**1985**). "Demonstration of adaptation in beluga whale echolocation signals," J. Acoust. Soc. Am. **77**(2), 726–730.

Belikov, R., and Bel'kovich, V. (**2003**). "Underwater vocalization of the beluga whales (*Delphinaterus leucas*) in a reproductive gathering in various behavioural situations," Oceanology **43**, 112–120; available at https://www.researchgate.net/publication/290528363_Underwater_vocalization_of_the_Beluga_Whales_Delphinaterus_leucas_in_a_reproductive_gathering_in_various_behavioural_situations.

Bergler, C., Schröter, H., Cheng, R., Barth, V., Weber, M., Nöth, E., Hofer, H., and Maier, A. (**2019**). "ORCA-SPOT: An automatic killer whale sound detection toolkit using deep learning," Sci. Rep. **9**, 10997.

Bergler, C., Smeele, S. Q., Tyndel, S. A., Barnhill, A., Ortiz, S. T., Kalan, A. K., Cheng, R. X., Brinkløv, S., Osiecka, A. N., Tougaard, J., Jakobsen, F., Wahlberg, M., Nöth, E., Maier, A., and Klump, B. C. (**2022**). "ANIMAL-SPOT enables animal-independent signal detection and classification using deep learning," Sci. Rep. **12**(1), 21966.

Bermant, P. C., Bronstein, M. M., Wood, R. J., Gero, S., and Gruber, D. F. (**2019**). "Deep machine learning techniques for the detection and classification of sperm whale bioacoustics," Sci. Rep. **9**(1), 12588.

Booy, K. V., Mouy, X., Ferguson, S. H., and Marcoux, M. (**2021**). "Spatio-temporal summer distribution of Cumberland Sound beluga whales (*Delphinapterus leucas*) in Clearwater Fiord, Nunavut, Canada," Arct. Sci. **7**(2), 394–412.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (**2020**). "End-to-end object detection with transformers," arXiv:2005.12872 [cs].

Castellote, M., Small, R., Lammers, M., Jenniges, J., Mondragon, J., Garner, C., Atkinson, S., Delevaux, J., Graham, R., and Westerholt, D. (**2020**). "Seasonal distribution and foraging occurrence of Cook Inlet beluga whales based on passive acoustic monitoring," Endang. Species Res. **41**, 225–243.

Chmelnitsky, E. G., and Ferguson, S. H. (**2012**). "Beluga whale, *Delphinapterus leucas*, vocalizations from the Churchill River, Manitoba, Canada," J. Acoust. Soc. Am. **131**(6), 4821–4835.

Erbe, C., and King, A. (**2008**). "Automatic detection of marine mammals using information entropy," J. Acoust. Soc. Am. **124**, 2833–2840.

Fish, M. P., and Mowbray, W. H. (**1962**). "Production of underwater sound by the white whale or beluga, *Delphinapterus leucas* (Pallas)," J. Mar. Res. **20**, 982; available at https://elischolar.library.yale.edu/journal_of_marine_research/982/.

Fisheries and Oceans Canada (**2012**). "Recovery strategy for the beluga whale (*Delphinapterus leucas*), St. Lawrence estuary population in Canada," technical report, https://www.sararegistry.gc.ca/virtual_sara/files/plans/rs_st_laur_beluga_0312_e.pdf (Last viewed September 16, 2024).

Garland, E. C., Castellote, M., and Berchok, C. L. (**2015**). "Beluga whale (*Delphinapterus leucas*) vocalizations and call classification from the eastern Beaufort Sea population," J. Acoust. Soc. Am. **137**(6), 3054–3067.

GitHub (**2024**). "End-to-end object detection with transformers," https://github.com/facebookresearch/detr (Last viewed September 16, 2024).

Goverment of Canada (**2014**). "Species at Risk Act Canada," technical report.

Hagiwara, M. (**2022**). "AVES: Animal vocalization encoder based on self-supervision," arXiv:2210.14493 [cs, eess].

Karlsen, J., Bisther, A., Lydersen, C., Haug, T., and Kovacs, K. (**2002**). "Summer vocalisations of adult male white whales (*Delphinapterus leucas*) in Svalbard, Norway," Polar Biol. **25**(11), 808–817.

Kowarski, K. A., and Moors–Murphy, H. (**2021**). "A review of big data analysis methods for baleen whale passive acoustic monitoring," Mar. Mammal Sci. **37**(2), 652–673.

Lesage, V., Barrette, C., Kingsley, M. C. S., and Sjare, B. (**1999**). "The effect of vessel noise on the vocal behavior of belugas in the St. Lawrence River Estuary, Canada," Mar. Mammal Sci. **15**(1), 65–84.

Miralles, R., Lara, G., Carrión, A., and Esteban, J. A. (**2013**). "Automatic detection and classification of beluga whale vocalizations," Adv. Appl. Acoust. **2**(2), 61–70, available at https://riunet.upv.es/handle/10251/62363.

Oedekoven, C., Marques, T., Harris, D., Thomas, L., Thode, A., Blackwell, S., Conrad, A., and Kim, K. (**2022**). "A comparison of three methods for estimating call densities of migrating bowhead whales using passive acoustic monitoring," Environ. Ecol. Stat. **29**, 101–125.

Panova, E. M., Belikov, R. A., Agafonov, A. V., and Bel'kovich, V. M. (**2012**). "The relationship between the behavioral activity and the underwater vocalization of the beluga whale (*Delphinapterus leucas*)," Oceanology **52**(1), 79–87, https://doi.org/10.1134/S000143701201016X.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (**2016**). "You only look once: Unified, real-time object detection" arXiv:1506.02640 [cs].

Roch, M. A., Lindenau, S., Singh Aurora, G., Frasier, K. E., Hildebrand, J. A., Glotin, H., and Baumann-Pickering, S. (**2021**). "Using context to train time-domain echolocation click detectors," J. Acoust. Soc. Am. **149**(5), 3301–3310.

Schölkopf, B., and Smola, A. J. (**2001**). "Support vector machines," in *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT Press, Cambridge, MA), pp. 187–188.

Schröter, H., Nöth, E., Maier, A., Cheng, R., Barth, V., and Bergler, C. (**2019**). "Segmentation, classification, and visualization of Orca calls using deep learning," in *ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Simard, Y., Roy, N., Giard, S., Gervaise, C., Conversano, M., and Ménard, N. (**2010**). "Estimating whale density from their whistling activity: Example with St. Lawrence beluga," Appl. Acoust. **71**(11), 1081–1086.

Sjare, B., and Smith, T. (**1986**). "The vocal repertoire of white whales, *Delphinapterus leucas*, summering in Cunningham Inlet, Northwest Territories," Can. J. Zool. **64**, 407–415.

Song, Z., Mooney, T. A., Quakenbush, L., Hobbs, R., Gaglione, E., Goertz, C., and Castellote, M. (**2023**). "Variability of echolocation clicks in beluga whales (*Delphinapterus leucas*) within shallow waters," Aquat. Mamm. **49**(1), 62–72.

Todd, N. R., Cronin, M., Luck, C., Bennison, A., Jessopp, M., and Kavanagh, A. S. (**2020**). "Using passive acoustic monitoring to investigate the occurrence of cetaceans in a protected marine area in northwest Ireland," Estuarine, Coastal Shelf Sci. **232**, 106509.

Ulloa, J. S., Haupert, S., Latorre, J., Aubin, T., and Sueur, J. (**2021**). "scikit-maad: An open-source and modular toolbox for quantitative soundscape analysis in Python," Methods Ecol. Evol. **12**, 2334–2340.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (**2023**). "Attention is all you need," arXiv:1706.03762 [cs].

Vergara, V., Michaud, R., and Barrett–Lennard, L. (**2010**). "What can captive whales tell us about their wild counterparts? Identification, usage, and ontogeny of contact calls in belugas (*Delphinapterus leucas*)," Int. J. Comparative Psychol. **23**, 278–309.

Vergara, V., and Mikus, M.-A. (**2018**). "Contact call diversity in natural beluga entrapments in an Arctic estuary: Preliminary evidence of vocal signatures in wild belugas," Mar. Mammal Sci. **35**, 434–465.

Vergara, V., Wood, J., Lesage, V., Ames, A., Mikus, M.-A., and Michaud, R. (**2021**). "Can you hear me? Impacts of underwater noise on communication space of adult, sub-adult and calf contact calls of endangered St. Lawrence belugas (*Delphinapterus leucas*)," Polar Res. **40**, 5521.

Zhong, M., Castellote, M., Dodhia, R., Lavista Ferres, J., Keogh, M., and Brewer, A. (**2020**). "Beluga whale acoustic signal classification using deep learning neural network models," J. Acoust. Soc. Am. **147**(3), 1834–1841.