TECHNOLOGY AND CODE
published: 09 August 2022
doi: 10.3389/fmars.2022.867857

Echofilter: A Deep Learning Segmention Model Improves the Automation, Standardization, and Timeliness for Post-Processing Echosounder Data in Tidal Energy Streams

OPEN ACCESS

Scott C. Lowe[1,2]***, Louise P. McGarry**[3]***, Jessica Douglas**[3]**, Jason Newport**[4,5]**, Sageev Oore**[1,2]**, Christopher Whidden**[1,4] **and Daniel J. Hasselman**[3]

[1]Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada, [2]Vector Institute, Toronto, ON, Canada, [3]Fundy Ocean Research Centre for Energy, Dartmouth, NS, Canada, [4]DeepSense, Dalhousie University, Halifax, NS, Canada, [5]Marine Environmental Research Infrastructure for Data Integration and Application Network, Halifax, NS, Canada

Edited by:
Wei-Bo Chen,
National Science and Technology Center for Disaster Reduction (NCDR), Taiwan

Reviewed by:
Matthew Lewis,
Bangor University, United Kingdom
Wen-Ray Su,
National Science and Technology Center for Disaster Reduction (NCDR), Taiwan

***Correspondence:**
Scott C. Lowe
scott.lowe@dal.ca
Louise P. McGarry
louise.mcgarry@fundyforce.ca

Specialty section:
This article was submitted to Ocean Solutions, a section of the journal Frontiers in Marine Science

Received: 02 February 2022
Accepted: 17 June 2022
Published: 09 August 2022

Citation:
Lowe SC, McGarry LP, Douglas J, Newport J, Oore S, Whidden C and Hasselman DJ (2022) Echofilter: A Deep Learning Segmention Model Improves the Automation, Standardization, and Timeliness for Post-Processing Echosounder Data in Tidal Energy Streams. Front. Mar. Sci. 9:867857. doi: 10.3389/fmars.2022.867857

Understanding the abundance and distribution of fish in tidal energy streams is important for assessing the risks presented by the introduction of tidal energy devices into the habitat. However, tidal current flows suitable for tidal energy development are often highly turbulent and entrain air into the water, complicating the interpretation of echosounder data. The portion of the water column contaminated by returns from entrained air must be excluded from data used for biological analyses. Application of a single algorithm to identify the depth-of-penetration of entrained air is insufficient for a boundary that is discontinuous, depth-dynamic, porous, and varies with tidal flow speed.

Using a case study at a tidal energy demonstration site in the Bay of Fundy, we describe the development and application of deep machine learning models with a U-Net based architecture that produce a pronounced and substantial improvement in the automated detection of the extent to which entrained air has penetrated the water column.

Our model, Echofilter, was found to be highly responsive to the dynamic range of turbulence conditions and sensitive to the fine-scale nuances in the boundary position, producing an entrained-air boundary line with an average error of 0.33 m on mobile downfacing and 0.5–1.0 m on stationary upfacing data, less than half that of existing algorithmic solutions. The model's overall annotations had a high level of agreement with the human segmentation, with an intersection-over-union score of 99% for mobile downfacing recordings and 92–95% for stationary upfacing recordings. This resulted in a 50% reduction in the time required for manual edits when compared to the time required to manually edit the line placement produced by the currently available algorithms. Because of the improved initial automated placement, the implementation of the models permits an increase in the standardization and repeatability of line placement.

Keywords: machine learning, deep learning, hydroacoustics, entrained air, marine renewable energy, tidal energy, environmental monitoring, marine technology

1 INTRODUCTION

The need for clean, non-carbon emitting, alternatives for power production is well established (IPCC, 2021). With advancements in technology, energy extraction from kinetic marine sources (ocean current, tidal energy streams, and wave) have recently emerged as potential contributions to the suite of renewable energies for the generation of electricity (Cada et al., 2007; Roberts et al., 2016; Copping et al., 2020; IRENA, 2020). In the case of energy extraction from tidal energy streams, tidal turbines are introduced into nearshore, coastal ecosystems that can be important habitats of major biological importance to fish for migration, nursery, and feeding activities (Blaber et al., 2000; Melvin and Cochrane, 2012; DFO, 2018; Tsitrin et al., 2022). The development of this nascent industry is therefore introducing new technologies and new but uncertain risks into the marine environment (DFO, 2008). In regions where fish stocks are managed or listed for special protections, regulators require monitoring for potential effects for fish as a condition for licensing tidal and other marine renewable energy (MRE) projects. Depending on local flow and bathymetric characteristics in the nearshore environments, the current flows suitable for tidal energy development can be turbulent (Cornett et al., 2015; Melvin and Cochrane, 2015; Williamson et al., 2017; Perez et al., 2021; Wolf et al., 2022), entraining persistent and deeply penetrating air into the water column. An efficient backscatterer of sound, the presence of air in the water column complicates the post-processing activities for data collected with acoustic instruments.

Hydroacoustic methods, applied to data collected with scientifically calibrated echosounders, are used to quantify the distribution and abundance of fish in the marine environment (Johannesson and Mitson, 1983; Fernandes et al., 2002; Benoit-Bird and Lawson, 2016). Echosounders emit a pulse of sound (a "ping") into the water and record the magnitude of the returned backscatter (the "echo") (Simmonds and MacLennan, 2005). The advantage of echosounders is the ability to sample the full water column in high spatiotemporal resolution. However, to achieve the goals of biological analyses for fish presence and distribution, backscatter recorded from physical interfaces must be excluded, including from the seafloor or sea surface (sea-air interface) and those portions of the water contaminated by backscatter from entrained air. The international standard solution to this is to use the software Echoview (Echoview Software Pty Ltd., Hobart, Australia), which enables advanced visualization and post-processing of hydroacoustic data. Echoview includes a library of highly configurable, parameterized algorithms by which to achieve the work of post-processing, including defining the boundaries of the region suitable for biological analyses.

The classical algorithms of Echoview generally produce appropriate placement for the lines designating the seafloor and sea surface given their continuous, strongly reflective, and non-porous natures. In contrast, the boundary of the entrained-air penetration is indistinct, porous, and discontinuous, and formed of local features that can only be distinguished from biological features through their broader context. The profile of entrained air is further complicated for recordings at sites where the penetration of entrained air is influenced by tidal flow speeds which can range from slack tide to 5 ms^{-1} (10 knots), e.g. Bay of Fundy (Karsten et al., 2011). These characteristics limit the potential for classical algorithms to successfully identify the extent of entrained air within the water column. This lack of automation has important consequences for hydroacoustic data post-processing and analyses:

- substantial and time-consuming manual edits are required to refine the ping-by-ping demarcation of the ambit of entrained air,
- the quantity of edits generates analyst fatigue putting at risk the regions where the full force of analyst attention is needed for discerning usable data, and
- standardization and/or repeatability is impossible to achieve between analysts and within the work of a single analyst.

Machine learning is a methodology that enables the construction of models through the use of example input/output data. In particular, deep learning allows us to build the complex models which are necessary to solve challenging tasks which would otherwise require a human to laboriously perform (LeCun et al., 2015; Schmidhuber, 2015; Goodfellow et al., 2016). Deep learning models have revolutionized computer vision over the last 10 years (Krizhevsky et al., 2012; He et al., 2016; Bengio et al., 2021), have been successfully applied to image segmentation tasks (Ronneberger et al., 2015; Redmon et al., 2016; Minaee et al., 2022), and have attained human-level or superhuman performance at narrow tasks (Karpathy, 2014; Russakovsky et al., 2015; He et al., 2016; Santoro et al., 2016; Silver et al., 2017). We hypothesised that a deep neural network would be able to solve the task of placing the entrained-air line correctly. Hence we deployed machine learning methods, with a convolutional neural network architecture inspired by U-Net (Ronneberger et al., 2015) and EfficientNet (Tan and Le, 2019), to determine whether such models can generate an entrained-air line with better placement than the existing classical algorithms (as implemented in Echoview) and thus reduce the amount of human labour needed to complete this task.

In the deep learning framework, an artificial neural network is instantiated with a particular architectural design (with randomly initialized parameters), and its parameters are iteratively updated through gradient descent in order to maximize performance at the objective task. Through this training process, the network learns to approximate a function that maps a set of input stimuli to the correct outputs. In the context of this work, the input to the model was a 2-D image-like representation of the hydroacoustic recording for which the axes are depth and time, and the intensity at each pixel is the volume backscattering strength (S_v dB re: 1 m^{-1}); we refer to this input as an *echogram*. The model's main output is a prediction of the depth of the entrained-air boundary line for each point in time (each ping). In addition to this, our model also predicts the depths of the seafloor and sea surface boundary lines, and (for each datapoint) whether the echosounder was active (emitting pings) or passive (in listening-only mode).

Our final implementation, *Echofilter*, is openly available under the AGPLv3 license. Python source code and a standalone Windows executable are available at https://github.com/DeepSenseCA/echofilter, with command line interface (CLI) and application programming interface (API) documentation available at https://DeepSenseCA.github.io/echofilter/.

2 MATERIALS AND EQUIPMENT

2.1 Data Sources

Hydroacoustic data was collected from two tidal energy demonstration sites within the Bay of Fundy in Nova Scotia, Canada: Minas Passage in which flow speeds can exceed 5 m s^{-1} (Karsten et al., 2011) and Grand Passage in which flow speeds can achieve 2.5 m s^{-1} (Guerra et al., 2021).

"Stationary" data was collected using a calibrated Simrad EK80 WBAT 7° split-beam echosounder operating in continuous wave (CW) mode at 120 kHz in Minas Passage and in Grand Passage. The echosounder, with its transducer in an upward facing orientation was attached to a platform deployed to the seafloor (see **Figure 1**). The seawater depth at the platform location varied with tide height from 29 m to 44 m at Minas Passage, and 14 m to 20 m at Grand Passage. The echosounder was deployed in Minas Passage for three 2-month periods in 2018. Data was recorded for 5 minutes every half hour. Passive data collection with the echosounder in listening-only mode to document system self-noise and record levels of ambient sound present at 120 kHz was collected during two of the three deployments. There were two deployments of the echosounder in Grand Passage during late 2019 and early 2020. In both cases, the echosounder was deployed for less than 14 days. Data collection cycle in Grand Passage consisted of one-hour continuous data collection in alternating hours. Short durations of passive data were collected each hour.

"Mobile" data was collected from the Minas Passage site using a calibrated Simrad EK80 WBT 7° split-beam echosounder operating in CW mode at 120 kHz. The transducer was deployed in a downward facing orientation attached via polemount to the vessel. The mobile survey pattern consisted of a set of six parallel transects, each of length 1.8 km and separated by 200 m, encompassing the Minas Passage multi-berth tidal energy demonstration site in the northern portion of the Passage, plus three reference transects located across the Passage near the southern shore. For ten of the seventeen mobile surveys, one additional transect was added to sample a region of interest in the demonstration site. The mobile surveys consisted of discrete 24-hour data collection periods during which the grid of transects was traversed four times, weather permitting. A completed grid consisted of one with-the-current and one against-the-current traverse of each transect. Seventeen such surveys were conducted between May 2016 and October 2018. Seawater depths ranged from 13 m to 67 m. Mobile data collection included periods of passive data collection with the transiting of each transect. No mobile data was collected in Grand Passage.

The echosounder data files were imported into Echoview (version 10.0) and post-processed in the typical way: (i) assigned calibration parameters, (ii) examined the data and removed noise, (iii) removed the passive data from further processing, (iv) set a line at constant range from the transducer face (1.7 m in this case) by which to exclude the transducer nearfield and, (v) applied Echoview algorithms to estimate, for each ping, the position of the seafloor (for downfacing echosounder) or sea surface (for upfacing echosounder) and the depth-of-penetration of the entrained air. In order to exclude the acoustic deadzone inherent in echosounder data (Simmonds and MacLennan, 2005), a one-meter offset was applied to the bounding line (seafloor or sea surface) and to the entrained-air line.

2.2 Data Partitioning

The full suite of Echoview files were divided into sets of files for training, validating, and testing the machine-learning models. The mobile downfacing dataset collected at Minas Passage consisted of 17 surveys, repeated at (a subset of) the same 10 transects on 17 different days spanning the course of three years. We selected two transects and placed all recordings from these in the test set. The remaining data was partitioned into training,

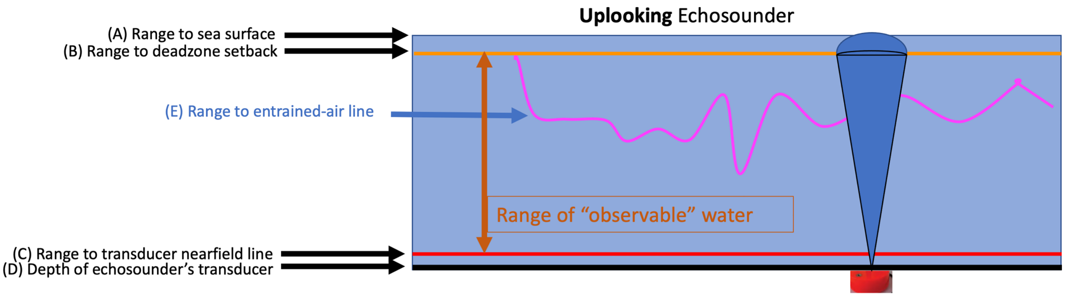
Uplooking Echosounder
(A) Range to sea surface
(B) Range to deadzone setback
(E) Range to entrained-air line
Range of "observable" water
(C) Range to transducer nearfield line
(D) Depth of echosounder's transducer

FIGURE 1 | Illustration of the boundaries defining the range of the observable water column for an upfacing echosounder. **(A)** Range to sea surface. **(B)** Deadzone setback: the range in which biomass within the acoustic beam is indistinguishable from the strong surface echo (Simmonds and MacLennan, 2005). **(C)** Nearfield line: a line set at a constant range to exclude the transducer "nearfield" in which the sound pulse is not yet organized (Simmonds and MacLennan, 2005). **(D)** Position of the upfacing transducer. **(E)** Entrained-air line: depth range of observable water is restricted to that not contaminated by entrained air.

validation, and (unused) "test2" partitions with an 80/10/10% split, stratified against the season in which the data was collected (winter vs non-winter) to ensure an equal split of the sparser winter recordings. The stationary data was grouped into blocks of 6 hours of consecutive recordings, and these blocks were partitioned at random (without stratification). We placed 80% of the MP:sta↑ and GP:sta↑ data files in the training partition. For the MP:sta↑ dataset, 10% of the data was used for model validation and 10% for final testing. Due to its smaller size, we did not use any GP:sta↑ data for the model validation process and kept the remaining 20% of the data for testing. The number of recordings and pings for each partition of each dataset is indicated in **Table 1**.

Files used for manual evaluation were selected from the MP:sta↑ and GP:sta↑ validation and test partitions, and chosen to ensure 24-hour coverage, with both neap tide and spring tide included. Stratifying the samples in this way ensured we would see examples of model performance under best-case and worst-case entrained-air scenarios. We also selected files from the MP:sta↑ and GP:sta↑ training partitions to inspect, in order to determine whether errors were due to applying the models on new data, indicating an issue with the model's ability to generalize (overfitting), or whether there was also a problem on the training data, indicating an issue with the model design (underfitting).

3 METHODS

3.1 Echoview Algorithm

As a baseline to benchmark our models against, we used Echoview algorithms to generate seafloor (downfacing recordings), sea surface (upfacing recordings), and entrained-air boundary lines.

Two separate Echoview algorithmic approaches were used for estimating the ambit of entrained air within the stationary and mobile datasets, respectively. For data collected during stationary surveys, periods of recorded passive-data were excluded and a 2-D Gaussian blur was applied to the remaining echogram using the Echoview "XxY Convolution" operator. We used a 13-by-13 kernel, and a standard deviation of $\sigma = 2.0$ in both depth (over return sample indices) and time (over ping indices) dimensions. We then used the "Threshold Offset" line picking operator, with a minimum threshold boundary of −80 dB, to search below the surface line and define the ambit of entrained air for each ping. The position identified within each ping was then used as the automated demarcation between entrained air and water column in the original S_v echogram.

For data collected during mobile surveys, the "Best Bottom Candidate" line picking operator was used to estimate the ambit of entrained air at each ping. Because the Best Bottom Candidate operator identifies the first instance of strong signals deeper than weak signals in the water column, we first inverted the intensity of the S_v echogram by multiplying the values by −1 and adding −150 to each result. We then used the Best Bottom Candidate operator, parameterized with −70 dB for the minimum S_v for a good pick and for the discrimination level, to identify the interface between entrained air ("weak") values, and water column ("strong") values in the inverse echogram. The position identified for each ping was then used as the automated demarcation between entrained air and water column in the original S_v echogram. This standardized protocol was used for the last 8 mobile surveys (surveys 10 through 17). For the first 9 mobile surveys, the protocol was inconsistent (sometimes including smoothing operations on the line, and offsets of varying sizes) and yielded variable outputs; these surveys were excluded from our benchmarking analysis described in Section 4.1.

The Best Bottom Candidate line picking operator was used to estimate the position of the seafloor (downfacing recordings), and the sea surface (upfacing recordings). For seafloor detection, the default settings were used and included a bottom offset of 0.5 m, except for the first two of the seventeen mobile surveys for which some of the parameters were adjusted. For sea surface detection, the default settings were used except for the backstep discrimination level which was halved to −25 dB.

The entrained-air and seafloor lines produced by the Echoview algorithms were used as seed lines which expert human annotators, with reference to the S_v echogram including a minimum S_v threshold set to −66 dB, then manually adjusted to create corrected, finalized annotations. These human-refined annotations were used as the targets for training the machine learning model.

3.2 Data Preprocessing

Annotated data was stored in Echoview EV files, which contain both the S_v data and human-generated annotations for the boundary lines. The EV files were opened in Python using win32com to interface with Echoview's programming interface (API), and exported into several files. The surface, seafloor, and entrained-air lines were exported into Echoview line (EVL) file format. The S_v data was exported into CSV format twice as follows. The first S_v CSV file ("raw S_v CSV") was exported with all EV exclusion settings disabled, and contained the entire S_v

TABLE 1 | Summary of datasets used in this study: their recording locations, mobility, and recording orientation.

				No. Recordings			No. Pings		
Dataset	**Location**	**Mobility**	**Orientation**	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
MP:mob↓	Minas Passage	Mobile	Downfacing (↓)	727	91	245	1.21M	148K	394K
MP:sta↑	Minas Passage	Stationary	Upfacing (↑)	7,249	919	875	2.45M	305K	300K
GP:sta↑	Grand Passage	Stationary	Upfacing (↑)	118	0	28	0.36M	0	96K

We indicate the sizes of the dataset partitions, in terms of the number of contiguous recordings (duration dependent on dataset), and the total duration of the recordings measured in number of pings (k, thousand; M, million).

data in the EV file. The second S_v CSV file ("clean S_v CSV") used the exclusion settings as implemented in the EV file so that all data which should be excluded from ecosystem analyses was masked out, appearing as the NaN indicator value -9.9×10^{37} in the output CSV. This means all datapoints above the entrained-air line, below the seafloor line (for downfacing recordings), passive data, bad data time periods where the analyst deemed a sequence of pings to contain data throughout the water column too contaminated by returns from entrained air or suspended sediment to use at all, and other miscellaneous localized "bad data" caused by anomalous events such as a rope drifting into view which the analyst had labelled for exclusion, were removed from the output (set to the NaN indicator value), leaving only the datapoints deemed as "good data" by the analyst.

Since this export process requires using Echoview to read in and export the data from the EV file, and Echoview is only available for Windows, this first step of the data processing pipeline must be performed on a Windows system with a licensed copy of Echoview installed. The remaining steps in the data processing and model training pipeline only require Python and can be run on any operating system.

The CSV files and EVL files were loaded into Python with a custom data loader. The depth resolution (and number of datapoints) per ping sometimes differed during a recording session, resulting in data with an uneven sampling resolution; we addressed this by finding the modal depth resolution across pings and linearly interpolating the data for each ping onto the same array of depth sample points. We created a "target mask" based on the location of NaN-values in the clean S_v CSV. This target mask corresponds to the overall target for the network's output. The depth lines loaded from the EVL files were linearly interpolated onto the same set of timestamps as the S_v data.

We observed some discrepancies between the depth lines and the mask, which was caused by (1) off-by-one differences when the line threshold is applied in Echoview compared with our own interpolation of the line; (2) analysts using boxes or freehand regions to annotate exclusion regions which are adjacent to the boundary lines. We handled this by identifying the upper and lower contiguous extent of the masked out area to generate new lines from the mask. For the entrained-air line, we primarily used the deepest extent of the two options as provided via the line annotation and the mask annotation. For the seafloor line, we primarily used the original line annotation as the network's output target, but we also produced a second line (with more aggressive removal) which extended higher up the water column to include any additional masked out area. The spare "aggressive" version of the seafloor line was included as an auxiliary target during training.

The surface line annotations were mostly unchanged by the annotators from the output produced by Echoview's algorithms. These were observed to be mostly accurate, but contained occasional large jumps in value. These outliers were detected and removed by using a median filter as follows. We applied a median filter with a kernel length of 201 and observed the residual between the raw signal and the median filter. Values more than 5 standard deviations (robustly estimated from the interquartile range, σ = iqr/1.35) were set to the median value. We then applied a median filter with a kernel length of 31 and removed anomalous values more than 4 standard deviations (robustly estimated from the interquartile range, σ = iqr/2.56) from the median. The second step was repeated until no anomalies were removed. Additionally, if the surface line was ever deeper than the entrained-air line, we set it to be the same depth as the entrained-air line. We found this anomaly removal process produced surface lines of sufficient quality. For downfacing samples, the surface line was set at 0 (coincident with the transducer face).

Passive data annotations were taken as hard-coded on/off cycles where known *a priori*. Otherwise, passive data collection periods were identified using a bespoke algorithm. The first S_v responses, corresponding to depths closest to the echosounder, have large intensities when the echosounder is active and much lower values when the echosounder is passive. We identified passive data periods by observing the first 38 depth sample points (after our interpolation step onto a common sampling grid). We took the difference in S_v between consecutive pings, and then the median across the first 38 depth samples for each ping. Median differences which exceeded ±25 dB were identified as boundary points between passive and active recording periods.

Bad data periods were identified as collections of consecutive pings for which all the data was masked out. Periods of passive data recording were excluded from the bad data periods. Bad data periods in which the entrained-air line was at or below the seafloor line throughout the entire period were also excluded.

Bad data patches were identified by the "pixels" in the echogram which were masked out for any reason not already covered by being above the entrained-air line, below the seafloor line, during a period of passive data collection, or during a period of time identified as a bad data period.

Our data was comprised of both upfacing and downfacing echosoundings. In the recording data structure, and exported CSV files, the y-dimension is stored as increasing distance from the echosounder. To standardize our inputs to the network, we flipped the orientation of the upfacing data such that increasing indices in the y-dimension corresponded to increasing depth within the water column.

The number of timepoints per file was much larger than we could reasonably supply to the network as a single input "image". Moreover, it is important that a single training batch contains a diversity of training data. To prevent the system from having to read in the contents of an entire recording file when needing to select only a small subset of the data to present for each training step, we broke the training data into chunks (shards) each with a length of 128 samples.

The pipeline for converting the CSV and EVL data into the preprocessed training shards can be executed with the command echofilter-generate-shards.

3.3 Training Inputs

When analysing echosounder data, it is common practice to offset the seafloor and entrained-air boundary lines by a fixed distance, 1 m for the echosounders used here. The purpose of the fixed-distance offsets are to exclude those portions of the data near boundaries, such as the sea surface or seafloor or the entrained-air boundary, that may be biased due to the echosounder deadzone (see **Figure 1**) which is a function of the shape of the spherically spreading beam intersecting with a surface (Simmonds and MacLennan, 2005). In addition, it generates a buffer between the boundary of the entrained air and the data reserved for biological analyses, so as to exclude returns from entrained air adjacent to, but not connected to, the pronounced entrained-air boundary. This ensures processing errs on the side of excluding slightly more data, instead of accidentally including bad data. Some datasets had an offset of 1 m included in the line definitions, whereas others did not. We standardized this by subtracting offsets from the lines which had them included. Consequently, the model's target output is to predict the exact boundary locations, and offsets can be added to its outputs as appropriate via optional settings in the Echofilter API.

Each training input image was normalized independently, based on the distribution of S_v values within the training input. Normalization was performed by subtracting the median over all S_v values, and dividing by a robust estimate of the standard deviation derived from the interdecile range (σ = iqr/2.56). A small number of NaN values were present in the raw S_v data, and these were set to a value of –3 after the normalization step.

The maximum apparent range of the echogram can in some cases be several times further than the actual depth of the water column. This is because the depth dimension corresponds to the time-of-flight of the signals, the maximum of which is determined by a maximum range parameter chosen by the operator of the echosounder, which may be held the same across many recordings and thus may be much larger than the local depth of the water column. In order to get the most precise output for the entrained-air lines from the trained model, we would like to zoom in on only the salient region of the image: the water column, extending from seafloor to sea surface. This allows the model to predict the boundary point with sufficiently high granularity. However, since the depth of the seafloor is not necessarily known *a priori*, the model needs to be able to determine the depth of the seafloor, or range to sea surface, from the full echogram as well. For testing, we thus use a two-step approach. First, the full echogram is presented to the network and the seafloor and/or surface lines are predicted. These outputs are used to zoom in on the water column. Second, this zoomed-in echogram is presented to the network, and precise seafloor, surface, and entrained-air lines are generated.

Inputs to the network are samples from the distribution of plausible echograms. During training, inputs to the network were drawn from the training partition and augmented with several operations. (i) Temporal stretching, stretch/squashed by a factor sampled log-uniformly from [0.5, 2]. (ii) Random depth cropping. With p = 0.1, the depth was left at the full, original extent. With p = 0.1, the echogram was zoomed in on the range from the shallowest surface depth to the deepest seafloor depth (the "optimal" zoom). With p = 0.4, the echogram was zoomed in to a random range of depths close to the optimal zoom, stretched or squashed by up to 25%, but never so much as to remove more than 25% of the entrained-air line or (for downfacing recordings) more than 50% of the seafloor line. With p = 0.4, the echogram was zoomed in to a random range of depths between the full original extent and the "optimal" extent. Depth upper and lower limits were selected uniformly across the appropriate range. (iii) Random reflection in the time (ping) dimension, performed with p = 0.5. (iv) "Color" jitter. We applied a "brightness" augmentation by offsetting normalized S_v values by a random additive offset chosen uniformly from [–0.5, +0.5], and a "contrast" augmentation by multiplying normalised S_v values by a random multiplicative factor chosen uniformly from [0.7, 1.3]. The same random offset and factor were used for each pixel in an echogram input. The order of the brightness and contrast augmentations was randomly selected for each input. (v) Elastic grid deformation, performed with p = 0.5. Elastic deformation was performed separately in the depth and time dimensions, to create an elastic grid deformation. We chose to deform the dimensions separately, instead of jointly as per a standard elastic deformation where space is stretched/squashed in a 2-D manner, because our targets are mostly at the ping level (depth of lines at each ping, whether the ping is passively or actively sampled, *etc.*) and apply to the entire column of data. A standard 2-D elastic deformation would break the relationship between our input and target; performing a joint elastic deformation on the echogram input would make it challenging to relate the input to the targets. We used σ = 8 in the time dimension, σ = 16 in the depth dimension, and α = 0.1 in both dimensions. The echogram was interpolated in 2-D, with the interpolation order randomly selected from linear, quadratic, and cubic (equal weighting).

Finally, the echogram was rescaled to size (128, 512) pixels (time-by-depth) for presentation to the network with nearest-neighbour interpolation.

3.4 Model Architecture

The model architecture used is a U-Net (Ronneberger et al., 2015) with EfficientNet MBConv blocks (Howard et al., 2017; Tan and Le, 2019), illustrated in **Figure 2**. This architecture is a convolutional neural network (CNN) with residual skip connections across blocks, 6 encoder layers where the size is spatially compressed, 6 decoder layers where the size is expanded back to the original input dimensions, and skip connections from the encoder to decoder blocks. The network has a backbone width of 32 channels throughout, and each MBConv block is inverse residual with an expansion factor of 6 (except the very first block, with has an expansion factor of 1). We used depthwise-separable convolutions with a kernel size of 5, and ReLU activations. We used Squeeze & Excite attention layers (Hu et al., 2019) on each block with a reduction factor of 2. In total, our final models each had 1.63M trainable parameters.

Since the input is rectangular, with higher resolution in the depth dimension, we downscaled the time dimension at a slower rate than the depth dimension. Downscaling was performed with max-pooling using a kernel size and stride of either 1×2 or 2×2 (alternating blocks). The depth dimension was downscaled after every block, whilst the time dimension was downscaled every other block.

The decoder branch was a mirror of the encoder: upscaling in the depth dimension after every block, and in the time dimension every other block. Upscaling was performed using bilinear interpolation with torch.nn.Upsample.

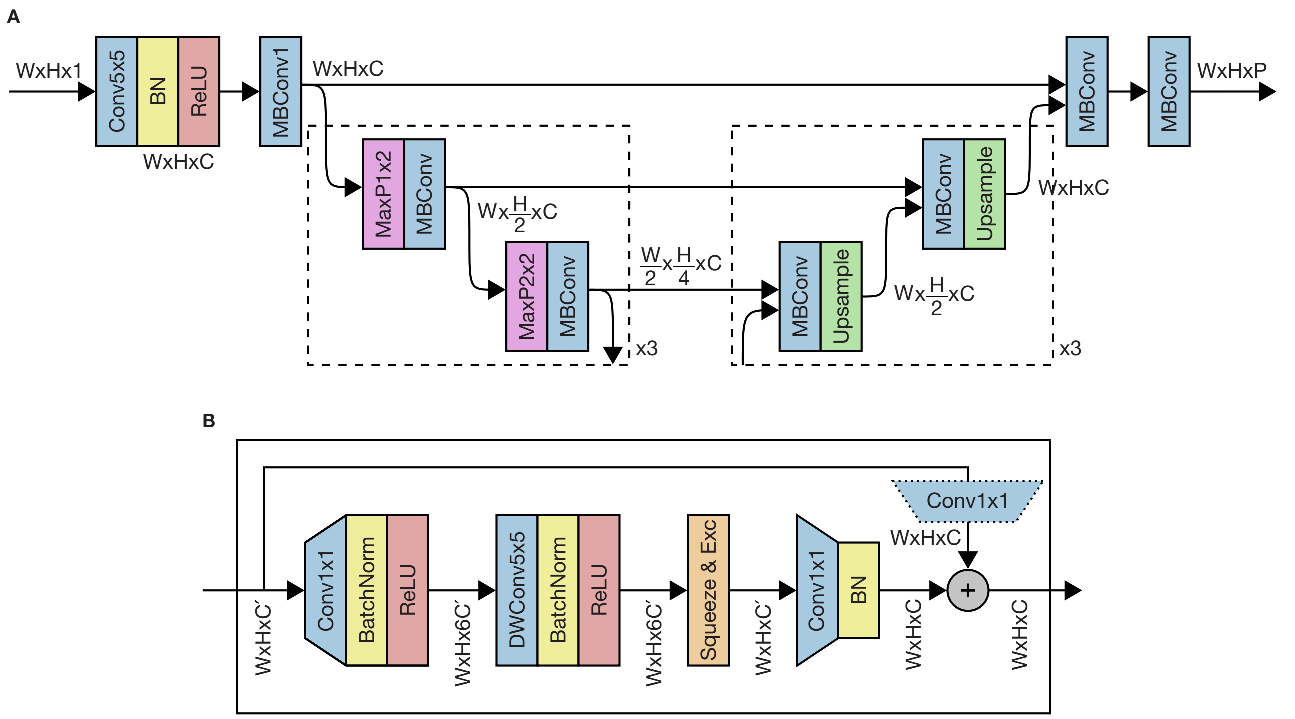

FIGURE 2 | Neural network architecture for the Echofilter model. **(A)** Main architecture, using an adaptation of the U-Net framework with 6 downsampling blocks, 6 upsampling units, and a single skip connection at each spatial resolution. The network contains a convolutional layer with 5×5 kernel and "same" padding (Conv5x5); BatchNorm (BN); rectified linear unit (ReLU); MBConv blocks (see panel **B**); max pooling (MaxP), with either 1×2 or 2×2 kernel and stride; and bilinear upscaling (Upsample) layers. The size of the latent representation of the image as it passes through the network is indicated. We train the network with W =128, H = 512, C = 32. **(B)** Structure of MBConv block, containing pointwise convolution (Conv1x1), depth-wise convolution (DWConv5x5) and Squeeze & Excite layers. For downscaling blocks C′ = C, and for upscaling blocks C′ = 2C. The pointwise convolution on the residual branch is only present for upscaling blocks, where C′ ≠ C.

The model has 10 output planes. These correspond to the probability (represented in logit form) that a pixel is at the boundary point for: (1) the entrained air (expanded), (2) the entrained air (original), (3) the seafloor line (expanded), (4) the seafloor line (original), (5) the surface line; and the probability (logit) that a pixel is within (6) a passive data period, (7) a bad data period (vertical region), (8) a miscellaneous bad data patch (to accompany expanded lines), (9) a miscellaneous bad data patch (to accompany original lines), (10) a miscellaneous bad data patch (to accompany original seafloor/expanded entrained air).

In practice, the expanded/original lines are almost identical and their pseudo-replication during training was superfluous, but their inclusion did indirectly increase the contribution of the entrained-air and seafloor lines towards the overall loss term. When performing inference with the model, we discard outputs 2, 4, 9, and 10.

For the Bifacing model, these 10 output planes are replicated three times. One is the standard output, the second are logits which are updated only on downfacing inputs, and the third are logits which are updated only on upfacing inputs. In this way, the model learns to represent conditional probabilities *P* (boundary | upfacing), *etc.* After training the model, we can ask it to predict the boundaries and masks agnostic of the orientation of the recording, or conditioned on the orientation (upfacing or downfacing).

3.5 Model Training

The model was optimized with gradient descent to minimize a loss function. The loss function acts as a proxy for the task of interest; a high loss corresponds to worse performance on the task, and a low loss to better performance. We constructed our loss function as the sum of several terms, each corresponding to one of the output planes produced by the model. The loss terms for the seafloor, sea surface, and entrained-air lines were each the cross-entropy between the column of logits across all depths for a single ping against a one-hot representation of the depth of the line. The loss terms for the passive collection and bad data periods were binary cross-entropy between the model's output for that ping (a single scalar, after collapsing the depth-dimension with log-avg-exp; Lowe et al., 2021) and the target value. The loss term for the localized bad data regions was binary cross-entropy. Outlying surface line values detected with our algorithm during preprocessing were masked out from the training objective. We took the mean over pings for all loss terms. We took the mean over the batch dimension; for outputs conditioned on the orientation of the echosounder, we masked out irrelevant samples before taking the batch-wise mean. When training the bifacing model with conditioning signals, all stimulus presentations were double-counted and the entire loss was divided by two to correct for this.

The model was optimized using the RangerVA optimizer (Wright, 2019), which combines RAdam, Lookahead, and gradient centralization (Zhang et al., 2019; Liu et al., 2020; Yong et al., 2020; Tong et al., 2022), with a weight decay of 1×10^{-5}. We used a batch size of 12 samples, and stratified the batches to contain the same ratio of downfacing and upfacing samples as available in the aggregated training set. The learning rate (LR) followed a cyclic learning rate schedule (Smith, 2015; Smith and Topin, 2017; Smith, 2018). In each cycle, the learning rate was warmed up for the first 10% of training, held constant for 40% of training, then warmed down for the last 50%. During the LR warmup period, the momentum was decreased from a maximum of $\beta_1 = 0.98$ to a base of $\beta_1 = 0.92$, and then increased back to 0.98 during the LR warmdown period. Both the LR and momentum were increased and decreased with cosine annealing. The second moment parameter was held constant at $\beta_2 = 0.999$ throughout training. In the first cycle, the model was trained for 100 epochs with a maximum learning rate of LR = 0.012. In subsequent cycles, the training duration was progressively doubled and maximum learning rate halved. We trained two models: the Bifacing model was trained for three cycles (700 epochs), whilst the Upfacing-only model was trained for two cycles (400 epochs). The model parameters were saved at the end of each cycle for subsequent analysis. We chose to stop the cyclic training process when the model's validation performance had reached a plateau.

The Upfacing model was trained on MP:sta↑and GP:sta↑ datasets, which contain only upfacing S_v recordings. The Bifacing model was trained on the MP:mob↓ dataset in addition to the MP:sta↑ and GP:sta↑ datasets. To address the smaller size of the GP:sta↑ dataset, we upsampled it by presenting echograms drawn from it twice per epoch instead of once (for both models).

The model architecture and training hyperparameters were each selected over a series of manual searches against the validation partition with short training durations of 5 or 20 epochs.

The network was trained using PyTorch 1.2.0 and CUDA 10.2. The model training and testing were done on the DeepSense high performance computing cluster with each training cycle or test using a 20 Core IBM Power8NVL 4.0 GHz compute node with 512 GB of RAM and a pair of NVIDIA Tesla P100 GPUs with 16 GB of GPU memory.

The Echofilter model can be trained using the command `echofilter-train`, with training parameters set at the command prompt.

3.6 Model Output Post-Processing

The neural network model is configured to generate predictions for each output type at the pixel level. That is to say, for each pixel in the input echogram, the network predicts a set of output variables at that particular pixel. For the passive data and bad data periods, we convert this 2-D output into a 1-D time series by taking the log-avg-exp over the depth dimension (Lowe et al., 2021).

We converted the model's output into lines as follows. For each boundary line, our model predicts the probability that each pixel is the location of said boundary. We integrated this probability across depth to create a cumulative probability density estimate, and identified the depth at which the cumulative probability exceeded 50%. In so doing, we generate a boundary depth prediction for every ping.

For the purposes of the machine learning model, all salient information needed to produce its outputs is contained in data at, or immediately surrounding, the water column. However, some echosound recordings have much greater range than this, extending out beyond the water column with a large number of samples. In order to put the echogram into the network, we scale the depth dimension down to 512 pixels. For echograms much larger than the water column, this step incurs a loss of information, since the water column may occupy only a small fraction of the 512 pixel resolution.

In order to alleviate this issue, the Echofilter protocol may run the echogram through the network twice, once zoomed out and once zoomed in on the water column. In the first instance, the echogram is "zoomed out" to the maximum extent and scaled down to 512 pixels. The depth of the seafloor or sea surface line is noted (the choice of line depending on echosounder orientation), and used to estimate the extent of the water column. Using a robust estimate of the standard deviation of depths in this line, we set our limit to be 4 standard deviations out from the mean of the line, or the furthest extent of the line, whichever is least distal. For upfacing recordings, we zoom in on the range from the deepest recording up to this depth minus an additional 2 m. For downfacing recordings, we zoom in on the range from the shallowest recording depth down to this depth plus an additional 2 m. After cropping the echogram down to this range of depths, we scale it down to 512 pixels and present it to the network again. The output from the second, "zoomed-in" presentation is used to determine the final entrained-air, surface/seafloor lines and other outputs.

This "zoom+repeat" technique provides gains (see Section 4.1.2), but we expect it to be needlessly expensive when only a small fraction of the echogram is outside the water column. For this reason, we only perform the second presentation if more than 35% of the echogram would be cropped out. This setting can be controlled with the `--autozoom-threshold` argument to Echofilter.

In our analysis, we observed that Echofilter's predictions of the locations of "bad data" were not sufficiently accurate. Furthermore, the mask can include a large number of small disconnected areas, which results in a inconveniently large number of regions to import into Echoview. In order to counter this, we can merge together regions with small gaps in between them, and impose a minimum size threshold on regions to be included in the output. We merged together consecutive passive regions annotations provided by the model with a gap smaller than 10 pings, and similarly for bad-data period labels. Any remaining regions shorter than 10 pings in length were omitted from the final output. For bad data patches, any patch with an area smaller than 25 ping-metres was omitted from the final output. In extremis, we can omit all bad-data annotations from Echofilter's region outputs.

An alternative solution to noisy outputs is to spatially smooth the output probabilities. We can apply a Gaussian smoothing kernel across each output plane before converting the logits into probabilities, and subsequently into lines and regions. However, we did not find this process yielded better results.

Lines and regions produced by Echofilter are exported into Echoview line (EVL) and region (EVR) files so they can be imported into Echoview. Additionally, the Echofilter command line supports saving lines and regions directly into the EV file which it is processing (Windows OS and a licensed copy of Echoview required), removing the subsequent step of manually importing the files.

Inference using a pretrained model can be performed on EV (Windows-only) or CSV files with the command `echofilter`. Pre-processing and post-processing options can configured be set at the command prompt.

3.7 Performance Quantification

To compare the quality of the outputs from the Echoview algorithm and our Echofilter models, we used a selection of metrics to quantify their performances. The results shown in Section 4.1 were determined by using these performance measurements on the test partition of each dataset. The test partition was neither seen by the model during training, nor used to optimize the model architecture and training process.

3.7.1 Intersection-Over-Union

The model's output was evaluated using the intersection-over-union score (IoU), also known as the Jaccard index metric (Jaccard, 1912), and Jaccard similarity coefficient score. This metric is commonly used to evaluate the performance of image segmentation models within the field of computer vision. The IoU of two masks is calculated by assessing their overlap; it is the ratio of the size of the intersection of the two masks against their union:

$$\mathrm{IoU}(\text{annotated}, \text{predicted}) = \frac{\text{Area (annotated} \cap \text{predicted)}}{\text{Area (annotated} \cup \text{predicted)}}. \tag{1}$$

For this study, one mask identifies the data marked as "good" by a human annotator, and the other mask is the data marked as "good" by the model. A higher IoU is better, indicating the two masks are better aligned. We chose to use this performance metric (instead of accuracy, *etc.*) because it is robust against padding the echogram with irrelevant range outside of the water column (below the seafloor for downfacing recordings, or above the sea surface for upfacing recordings).

For the MP:sta↑ and GP:sta↑ datasets, the IoU measurements we report are the total area of the mask intersections across the whole test set, divided by the total area of the union of the two masks (i.e. the division operation performed after the summation). For the MP:mob↓ dataset, the IoU reported is the average IoU over all the EV files in the test set (i.e. the division operation performed before the mean). In both cases, we determine the standard error (SEM) by considering the distribution of IoU scores over EV files. For any recording where the target mask is all marked as False (no good data), the intersection of the predicted area with the target area is always 0, and any prediction from the model results in a anomalously minimal score. Consequently, we excluded examples where the target was an empty mask when measuring the SEM.

3.7.2 Mean Absolute Error

We performed further evaluation of the model's outputs using the mean absolute error (MAE) performance metric. The MAE is defined as

$$\mathrm{MAE} = \frac{1}{n}\sum_{\forall i} \left| y_i - \hat{y}_i \right|, \tag{2}$$

where y_i is the target value for the i-th ping, $\hat{y}_i$ is the predicted value generated by the model, and n is the number of pings to average over. We applied the MAE to measure the quality of the output lines. In this context, the MAE corresponds to the average distance (across pings) of the model's line from the target line. A smaller MAE is better, indicating the model's line is (on average) closer to the target line.

When measuring the MAE of the lines, we excluded pings which were marked as being within a passive or bad data region in the target annotations. To find the overall MAE, for MP:mob↓ we determined the MAE within each file and averaged over files; for MP:sta↑ and GP:sta↑ we averaged over all pings across all files in the test set, weighting each ping equally.

Additionally, we report the standard error of the MAE. This is determined by computing the MAE for each test file, and measuring the standard error across these independent measurements.

3.7.3 Root-Mean-Square Error

We measured the root-mean-square error (RMSE) in a similar manner to the MAE. The RMSE is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{\forall i} (y_i - \hat{y}_i)^2}, \tag{3}$$

where y_i is the target value for the i-th ping, $\hat{y}_i$ is the predicted value generated by the model, and n is the number of pings to average over.

When measuring the overall RMSE on the test data, we first found the average MSE over the whole test set, then took the square root. Sample/file weighting, and standard error determination, was performed the same way as for the MAE.

3.7.4 Cumulative Error Distribution

We were particularly interested in how much human labour would be saved by the improvement in the annotations. There clearly must exist some error threshold below which errors in the annotation have no significant impact on downstream analysis and hence do not need to be fixed by the analyst. We speculate that this error tolerance threshold may be at around 0.5 m to 1.0 m, since the lines are offset by 1.0 m before performing biological analyses to ensure all “bad data” is excluded. Hence we can crudely estimate what fraction of the model’s output needs to be adjusted by the analyst by considering what fraction of pings are within 0.5m or 1.0m of the target line.

Since we can not be sure what the appropriate error tolerance is — and the tolerable error threshold may vary depending on the application — we can evaluate the performance of the model over a range of potential tolerance values by considering the cumulative distribution of the absolute error. Such a plot shows the fraction of outputs which are within a certain absolute error threshold, and is similar to a receiver operating characteristic (ROC) curve. If we seek to optimize this curve without assigning any particular error tolerance threshold, we can consider the total area above the curve (the expected rejection rate over all error tolerance thresholds), which we seek to minimise. This area is precisely equal to the MAE metric.

3.7.5 Test Data Weighting

For the MP:mob↓ dataset, test recordings were taken from two held out transects — no recordings from these transects were presented during training. This allows us to evaluate the performance of the model at novel recording locations which the model has not seen before. Unfortunately, the protocol for annotating the seafloor was not consistent for the first 9 of the 17 MP:mob↓ surveys; hence we evaluated the seafloor line and overall IoU only on test data from the final 8 surveys. The entrained air, passive data collection, bad data periods, and bad data patches were evaluated on all 17 of the MP:mob↓ surveys.

For the MP:sta↑ and GP:sta↑ datasets, our target surface lines were generated with the Echoview surface line detector, followed by automated anomaly detection, as described in Section 3.2. However, in some cases the Echoview algorithm fatally failed to detect the water–surface boundary, placing the surface line impossibly close to the echosounder, or impossibly far away. When evaluating the surface line on the test set, we dropped recordings where the “target” surface line depth was outside the known range of low to high tide water depths for that recording site (Minas Passage: 28.5 m to 44 m; Grand Passage: 13.5m to 20 m). This allowed us to evaluate the Echofilter model’s surface line predictions against sane target values, but this selectively removed almost all the occasions where the Echoview algorithm’s predictions were wrong, severely compromising our ability to evaluate the performance of the Echoview algorithm at generating the surface line.

4 RESULTS

We measured the performance of our models using coarse-grained quantitative metrics (Section 4.1), and compared to the output of algorithms built into Echoview as a baseline. To further contextualise the level of performance attained by our models, we measured the level of agreement between expert annotators separately annotating the data (Section 4.2). We also evaluated the performance by detailed investigation with qualitative outputs (Section 4.3), and finally we evaluated the practical output of the model by measuring the amount of time taken to audit and correct the model output (Section 4.4).

4.1 Quantitative Evaluation

We evaluated the overall performance of our final model by comparing the final “good data” mask produced by the model with that of the target labels. The target mask indicates which values within the echogram should be included in biological analyses. This mask excludes all values above the entrained-air line, below the seafloor line, during passive data collection regions, or marked as “bad data”. Our model produces outputs corresponding to each of these elements, and combining these outputs allows us to generate a final output mask. We measured the alignment between the two masks using the Intersection-over-Union (IoU), described in Section 3.7.1.

For our purposes, the most important output from the Echofilter model was the entrained-air line, which provides segmentation between the air entrained into the water column, and the rest of the water column. To provide a human-interpretable measurement of the error in the placement of this line, we measured the mean absolute error (MAE) and root-mean-square error (RMSE) between the depth of model’s entrained-air line and the target. See Section 3.7.2 and Section 3.7.3 for more details.

Other outputs from the model were evaluated similarly, using the IoU, MAE, and/or RMSE. In all cases, we show the performance of the model on the test partition, which was held out during all stages of model development and training.

4.1.1 Performance Break-Down Across Outputs

We investigated the performance of the final Upfacing (@400 epoch) and Bifacing (@700 epoch) models across all outputs produced by the network, and compared the quality of these outputs against the Echoview algorithm. The results were evaluated against the target annotations produced by a human expert, except for the surface line where the target was taken from the line produced by the Echoview algorithm but with anomalous values rejected (see Section 3.7.5).

We considered the IoU for each output, the results for which are shown in **Table 2**. The overall IoU compares the overall mask produced by removing pixels above the entrained-air line, below the seafloor line (if downfacing), during periods marked as passive data collection, and bad data annotations; we compare the mask obtained with the model against that from human annotation. For the entrained-air line, the IoU measurement considers the area beneath the entrained-air line — for upfacing recordings this extends to the echosounder, and for downfacing recordings it extends to the seafloor line provided by the expert’s annotation. Similarly the IoU measurement for the surface line extends from the surface to the echosounder, and is only measured for upfacing recordings. For the seafloor line, we compare the area from the seafloor line to the echosounder. For passive data region annotations, we compare the set of pings identified as passive by the model with the target annotations, performing a 1-D IoU calculation. The vertical bad data periods are measured in the same way as the passive data region annotations, using

TABLE 2 | Final model performance (agreement with manual annotation) for each output.

	MP:mob↓		MP:sta↑			GP:sta↑		
Output	**Echoview**	**Echofilter Bifacing**	**Echoview**	**Echofilter Upfacing**	**Echofilter Bifacing**	**Echoview**	**Echofilter Upfacing**	**Echofilter Bifacing**
	Intersection-over-Union (%; larger is better)							
Overall	96.80 ± 0.34	***99.15*** ± 0.08	90.41 ± 0.76	***95.08*** ± 0.34	*94.91* ± 0.35	87.66 ± 1.05	*92.10* ± 1.00	***92.97*** ± 1.00
Entrained-air	97.37 ± 0.31	***99.11*** ± 0.09	91.63 ± 0.72	***96.05*** ± 0.29	95.96 ± 0.28	89.06 ± 1.03	*94.49* ± 0.50	***94.95*** ± 0.29
Surface	–	–	***99.83*** ± 0.05	*99.82* ± 0.02	**99.83** ± 0.02	–	*98.59* ± 1.22	***99.86*** ± 0.01
Seafloor	99.33 ± 0.08	***99.79*** ± 0.03	–	–		–	–	–
Air–Seafloor	96.81 ± 0.34	***99.16*** ± 0.08	–	–		–	–	–
Passive	–	99.78 ± 0.06	–	***100.0*** ± 0.00	***100.0*** ± 0.00	–	*99.97* ± 0.01	***100.00*** ± 0.00
Bad data period	–	–	–	***40.58*** ± 7.64	*38.92* ± 7.42	–	*24.68* ± 7.65	***25.78*** ± 8.07
Patch (anomaly)	–	0.00 ± 0.00	–	***0.30*** ± 0.12	***0.30*** ± 0.11	–	***0.20*** ± 0.07	***0.20*** ± 0.07
	Mean Absolute Error (m; smaller is better)							
Entrained-air	1.178 ± 0.295	***0.325*** ± 0.031	2.187 ± 0.147	***0.981*** ± 0.044	1.005 ± 0.045	1.252 ± 0.198	*0.577* ± 0.074	***0.532*** ± 0.031
Surface	–	–	***0.062*** ± 0.018	*0.063* ± 0.007	**0.062** ± 0.007		*0.235* ± 0.232	***0.024*** ± 0.002
Seafloor	0.279 ± 0.032	***0.089*** ± 0.012	–	–	–	–	–	–
	Root-Mean-Square Error (m; smaller is better)							
Entrained-air	6.436 ± 0.390	***1.281*** ± 0.064	4.275 ± 0.196	***2.181*** ± 0.085	2.228 ± 0.088	2.995 ± 0.508	*1.244* ± 0.137	***1.104*** ± 0.060
Surface	–	–	1.323 ± 0.140	0.149 ± 0.020	***0.134*** ± 0.019	–	*3.019* ± 1.073	***0.035*** ± 0.003
Seafloor	2.017 ± 0.187	***0.292*** ± 0.024	–	–	–	–	–	–
	Proportion of pings where line placed within 0.5 m of target (%; larger is better)							
Entrained-air	71.24 ± 1.60	***88.30*** ± 0.82	52.67 ± 2.09	**61.28** ± 1.26	*61.11* ± 1.27	61.18 ± 2.58	*69.85* ± 2.54	***70.17*** ± 1.47
Surface	–	–	99.39 ± 0.15	*99.67* ± 0.27	**99.68** ± 0.26	–	*99.44* ± 0.57	***99.98*** ± 0.01
Seafloor	89.27 ± 1.02	***97.26*** ± 0.70	–	–	–	–	–	–
	Proportion of pings where line placed within 1.0 m of target (%; larger is better)							
Entrained-air	80.20 ± 1.33	***93.09*** ± 0.59	58.34 ± 1.99	***74.76*** ± 0.97	74.33 ± 1.01	69.10 ± 2.70	*83.70* ± 1.84	***84.93*** ± 0.92
Surface	–	–	99.44 ± 0.15	*99.87* ± 0.15	**99.88** ± 0.15	–	*99.46* ± 0.57	***100.00*** ± 0.00
Seafloor	95.30 ± 0.49	***98.75*** ± 0.53	–	–	–	–	–	–
	Proportion of pings where line placed within 2.0 m of target (%; larger is better)							
Entrained-air	87.58 ± 1.01	***96.46*** ± 0.38	68.08 ± 1.80	***86.50*** ± 0.66	86.12 ± *0.69*	80.61 ± 2.26	*93.33* ± 1.17	***94.20*** ± 0.45
Surface	–	–	99.52 ± 0.15	99.93 ± 0.11	***99.94*** ± 0.10	–	*99.46* ± 0.57	***100.00*** ± 0.00
Seafloor	98.48 ± 0.18	***99.78*** ± 0.16	–	–	–	–	–	–

The performance of the final Upfacing (@400 epoch) and Bifacing (@700 epoch) models, with thresholded zoom+repeat, merging/ignoring small output regions; compared against the performance of the Echoview algorithm as a baseline. Bold: best model. Italic: no significant difference from best (two-sided Wilcoxon signed-rank test, p>0.05).

a 1-D IoU. The IoU for the bad data patches is a comparison of the area marked as bad data by the model with a target mask indicating the locations of bad data patches.

We found the entrained-air and seafloor boundaries produced by both models had statistically significantly higher agreement with the human annotation than the lines produced by the Echoview algorithm (two-sided Wilcoxon signed-rank test, p<0.05). There was no significant difference between the outputs from the two models, except from the entrained-air line on MP:sta↑ where the magnitude of the difference in performance was small. Correspondingly, the quality of the overall output of the models were not significantly different from each other, but were significantly better than the Echoview algorithm.

The passive region annotations are highly accurate, reaching 100% accuracy on MP:sta↑ and GP:sta↑. On the MP:mob↓ dataset, the Bifacing model attains an IoU of 99.78%.

The bad data period annotations were challenging for the model to replicate, attaining an IoU of only 40% on MP:sta↑ and 25% on GP:sta↑. The anomalous bad data patches were impossible for the network to learn with any meaningful reliability, with an IoU of ≤0.3%. The poor performance of both of these annotations yields an increase in performance when small outputs are ignored (as seen in Section 4.1.2). On GP:sta↑, the bad data period annotations are sufficiently poor to yield an increase in performance when they are dropped entirely (see Section 4.1.2).

We measured the mean absolute error (MAE) of the entrained-air, surface, and seafloor lines (described in Section 3.7.2). As shown in **Table 2**, we found that the Bifacing model placed the entrained-air line on MP:mob↓ with only 0.325 m average error — a significant reduction (two-sided Wilcoxon signed-rank test, p<0.05) over the Echoview algorithm, which had over three times as much error on average. For the MP:sta↑ and GP:sta↑ datasets, the two Echofilter models had comparable performance, a significant reduction in error against the Echoview algorithm baseline which had more than twice the error of Echofilter on both the upfacing, stationary datasets. Our findings when evaluating the entrained-air lines using the RMSE metric, and the proportion of pings within fixed distances of the target lines, were the same as with MAE.

The larger absolute error on the GP:sta↑ dataset (0.53 m to 0.58 m) and MP:sta↑ dataset (1.0 m) is indicative of the increased difficulty intrinsic to these datasets collected at sites where the persistence, depth-of-penetration, variability of depth-of-penetration, and proportion of water column contaminated by entrained air exceeded that typically found at the transect locations sampled by the MP:mob↓ surveys. In particular, we note that the average and standard deviation of the depth-of-penetration for the entrained air was (12.4 ± 4.8) m for MP:sta↑, (6.7 ± 2.1) m for GP:sta↑, but only (2.4 ± 2.2) m for MP:mob↓. This corresponds to (34 ± 13) % of the water column for MP:sta↑, (41 ± 12) % for GP:sta↑, but only (6.2 ± 5.3) % for MP:mob↓.

As shown in **Figures 3A–C**, the cumulative error distribution curve produced by Echoview is dominated by Echofilter for all

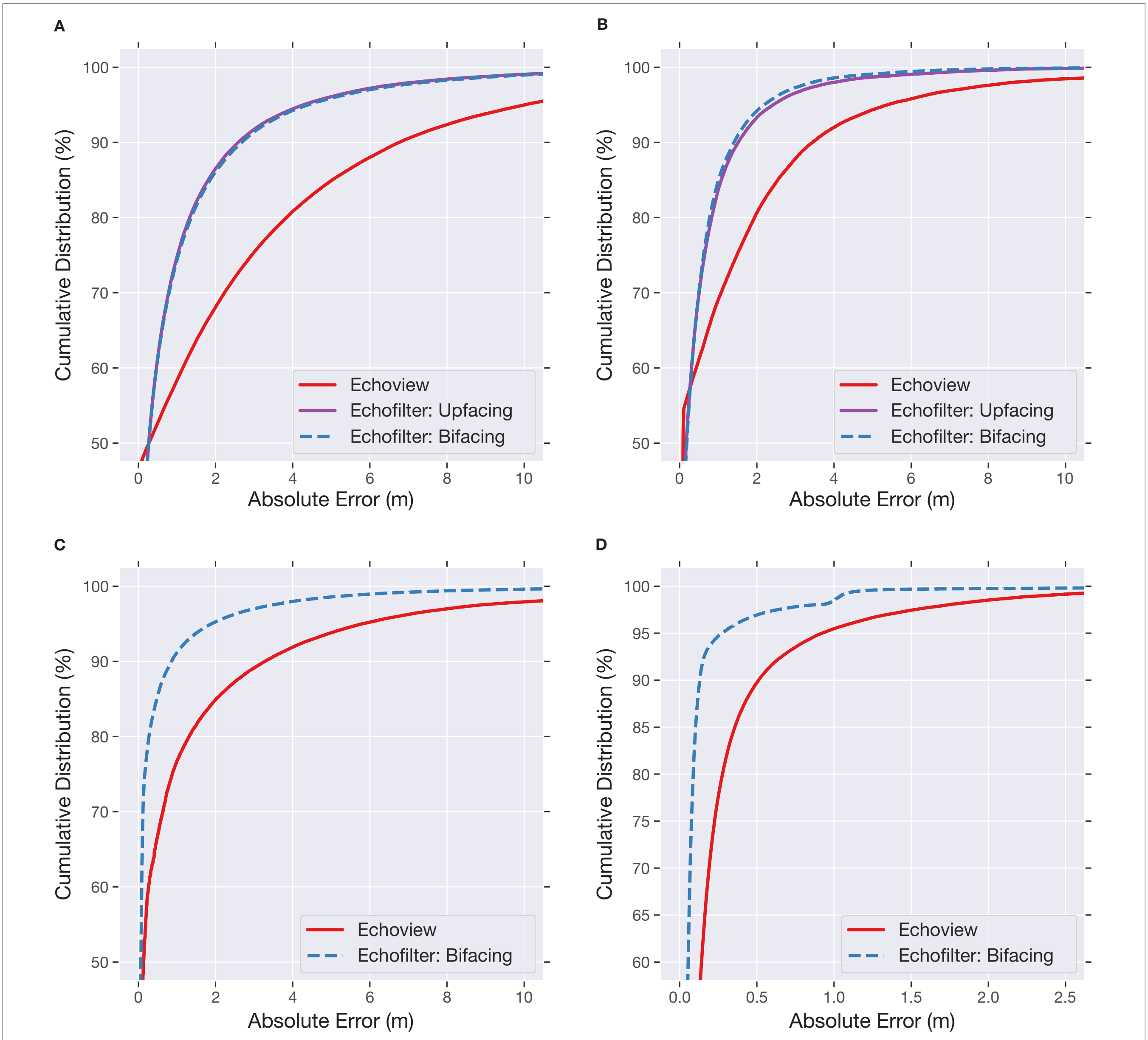

FIGURE 3 | Cumulative distribution for the absolute error of entrained-air **(A–C)** and seafloor **(D)** lines generated by the models (Echoview: red; Upfacing@400ep: magenta; Bifacing@700ep: blue). This indicates (on the y-axis) the fraction of pings where the output line was within a given threshold distance (x-axis) of the target line. **(A)** Error in entrained-air line on MP:sta↑. **(B)** Error in entrained-air line on GP:sta↑. **(C)** Error in entrained-air line on MP:mob↓. **(D)** Error in seafloor line on MP:mob↓.

values in the range of interest. Echoview appears to outperform Echofilter when using very narrow error thresholds (error < 0.2 m; the easiest 50% of the data), however this is an artifact of the manual data annotation process, in which Echoview was used to generate initial annotations which were then corrected as needed by a human expert.

As shown in **Table 2**, we found that the Bifacing model placed the seafloor line with very low error (only 0.09 m on average) on the MP:mob↓ test set. Again, this was significantly lower than the Echoview algorithm, for which the MAE was three times higher (0.28 m). Similar results were seen with the RMSE and proportion of pings within a tolerance threshold. As shown in **Figure 3**, the Echofilter cumulative error distribution for the seafloor line dominates Echoview. We note there is a step jump in Echofilter model performance at 1.0 m error; this is caused by inconsistencies in the training data annotation in the first 9 surveys which impacted the model fit (evaluated only on the last 8 surveys).

For the surface line annotation, we find that the Echofilter models have a MAE comparable to Echoview on MP:sta↑ and outperform Echoview when considering the RMSE and fraction of pings within 0.5 m to 2.0 m error threshold. This is because the Echofilter models do not produce the anomalous surface line depths seen with the Echoview line generation, which we removed from our training and target lines. On GP:sta↑, the Bifacing model produced better surface lines than the Upfacing model. Manual inspection of the results demonstrates that the Upfacing model is sometimes confused by reflections in the additional range of these recordings (following the erroneous training targets generated by Echoview), whilst the Bifacing model was not confused by these reflections.

4.1.2 Impact of Post-Processing Steps

We evaluated the performance of the final Upfacing and Bifacing models before and after each post-processing step impact described in Section 3.6. Our results are shown in **Table 3**.

Compared with applying the model only once on the full echogram, using the two step "zoom+repeat" stimulus presentation provided a statistically significant increase in the entrained-air line placement as evaluated by the MAE (two-sided Wilcoxon signed-rank test, $p<0.05$) on all datasets, and for both the Upfacing and Bifacing model. The overall IoU also significantly increased, except for the Bifacing model on MP:sta↑ where "zoom+repeat" caused a very small, but statistically significant, decrease.

We also considered using a threshold of 0.35 to determine when to do the zoom+repeat step, following our assumption that a second application of the model on a zoomed-in echogram is not necessary when less than 35% of the echogram data is outside the surface–seafloor extent. We found that using this threshold had no impact on the performance of the models on the MP:mob↓ and GP:sta↑ datasets, where the range of the data extended far outside the surface–seafloor extent and hence zoom+repeat was *de facto* always applied. On the MP:sta↑ dataset, where the recording range was not much further than the distance from seafloor to sea surface, there was a significant decrease in performance when a threshold was used to determine when to apply a second round of the model. This suggests that the zoom+repeat protocol should always be used in order to yield the best annotation with the model. Nonetheless, the rest of our results present in this paper use the (faster to perform) thresholded zoom+repeat, with a threshold of 0.35.

The remaining optional post-processing steps were considered with thresholded zoom+repeat in place. We found no significant differences in the overall IoU when small regions were merged together or dropped from the output (changing the way regions are handled has no effect on the entrained-air line placement). Omitting bad data regions and patches entirely had a positive impact on the overall performance on the GP:sta↑ data, but a negative impact on MP:sta↑ data. This was because the bad data period predictions (as seen in **Table 2**) were notably worse on GP:sta↑ than MP:sta↑. There was no impact on MP:mob↓ data because the model did not predict any bad data regions on this test dataset.

TABLE 3 | Impact of post-processing steps on the model performance metrics.

	Overall IoU (%)			Entrained-air MAE (m)		
Model	**MP:mob↓**	**MP:sta↑**	**GP:sta↑**	**MP:mob↓**	**MP:sta↑**	**GP:sta↑**
Echoview algorithm	96.80 ± 0.34	90.41 ± 0.76	87.66 ± 1.05	1.178 ± 0.295	2.187 ± 0.147	1.252 ± 0.198
Upfacing w/o zoom	–	95.06 ± 0.34	88.06 ± 3.88	–	0.987 ± 0.045	0.629 ± 0.076
w/zoom+repeat	–	***95.11*** ± 0.35	*92.09* ± 1.01	–	***0.950*** ± 0.041	***0.574*** ± 0.071
w/thresholded z+r	–	94.27 ± 0.46	*92.07* ± 1.01	–	0.981 ± 0.044	*0.577* ± 0.074
+ ignore small regions	–	95.08 ± 0.34	*92.10* ± 1.00	–	0.981 ± 0.044	*0.577* ± 0.074
+ ignore all "bad data"	–	94.77 ± 0.44	***93.01*** ± 0.76	–	0.981 ± 0.044	*0.577* ± 0.074
+ logit smoothing	–	94.27 ± 0.46	*92.48* ± 0.86	–	1.099 ± 0.046	*0.623* ± 0.095
Bifacing w/o zoom	98.59 ± 0.09	***94.90*** ± 0.35	88.35 ± 3.93	0.402 ± 0.030	1.004 ± 0.045	0.589 ± 0.047
w/zoom+repeat	***99.16*** ± 0.08	94.86 ± 0.40	*92.95* ± 1.01	***0.325*** ± 0.031	***0.979*** ± 0.044	***0.532*** ± 0.031
w/thresholded z+r	***99.16*** ± 0.08	***94.90*** ± 0.35	*92.95* ± 1.01	***0.325*** ± 0.031	1.005 ± 0.045	***0.532*** ± 0.031
+ ignore small regions	*99.15* ± 0.08	***94.91*** ± 0.35	*92.97* ± 1.00	***0.325*** ± 0.031	1.005 ± 0.045	***0.532*** ± 0.031
+ ignore all "bad data"	*99.15* ± 0.08	*94.74* ± 0.42	***93.45*** ± 0.64	***0.325*** ± 0.031	1.005 ± 0.045	***0.532*** ± 0.031
+ logit smoothing	98.90 ± 0.08	94.35 ± 0.43	*93.12* ± 0.67	0.385 ± 0.030	1.103 ± 0.051	*0.570* ± 0.047

Bold: best pre-processing option. Italic: no significant difference from best (two-sided Wilcoxon signed-rank test, p>0.05).

We considered the effect of logit smoothing on the model's final output by applying this postprocessing step, in addition to thresholded zoom+repeat and ignoring all bad data annotations, with a Gaussian kernel size of 1. We found that logit smoothing had a significant negative impact on the accuracy of the entrained-air line placement, and on the overall mask output, for all datasets.

4.1.3 Impact of Model Training Duration

We investigated the impact of training time on the final model outputs. We compared the output of each of the models at the end of each stage of the cyclic training process. For this analysis, we used thresholded zoom+repeat, and merged/ignored small regions in the model output.

As shown in **Table 4**, we found that further training cycles improved the performance on MP:sta↑ and MP:mob↓, though with diminishing returns. Additional training *reduced* the performance on GP:sta↑, but the reduction was not statistically significant.

4.2 Inter-Annotator Agreement Benchmarking

The extent to which air is entrained in the water column is not observed directly, and can only be estimated based on the echosounder recordings. With training and experience, human annotators can learn which datapoints correspond to entrained air and which to fish populations within the water column. However, without a ground truth measurement, the annotations are subjective and will differ between annotators.

With this in mind, it is difficult to know how well we could expect an ideal model to perform at the task. It is infeasible to expect perfect agreement between the model and the human annotations, since human annotators do not always agree amongst each other and are not necessarily consistent in their choice of line placement. We endeavoured to quantify how well our model performs by measuring the agreement between two human annotators, which acts as a baseline to estimate the Bayes error rate.

We selected 10 EV files from the Grand Passage stationary-upfacing dataset (GP:sta↑), ensuring that the selected files were composed of sufficiently complex data so that any differences in line placement between each annotator would be highlighted. Annotations were generated by Echoview using the preexisting workflow. The Echoview annotations were edited independently by JD and LPM in order to create two sets of finalized annotations for all 10 files. We then created annotations using Echofilter (models Upfacing@400ep and Bifacing@700ep, using thresholded zoom+repeat, and dropping small regions). While both annotators are experts in this field, JD was the most experienced at handling this data — her annotations constituted the majority of the annotations used to train the models. Consequently, we treated JD's annotations as the ground truth labels, and measured the performance of the other annotation methods in comparison to her labels.

As shown in **Table 5**, we found that the level of agreement in placement of the entrained-air line between Echofilter and JD exceeded that of LPM, with higher IoU and a smaller average distance between the line depths, though the difference was not statistically significant (p>0.05). This suggests our model outputs have an accuracy comparable to human-level performance at this task.

TABLE 4 | Performance of models after each training cycle (different total training durations).

		Overall IoU (%)			Entrained-air MAE (m)	
Model	**MP:mob↓**	**MP:sta↑**	**GP:sta↑**	**MP:mob↓**	**MP:sta↑**	**GP:sta↑**
Upfacing 100ep	–	*95.05* ± 0.33	***93.32*** ± 0.87	–	1.001 ± 0.046	***0.520*** ± 0.033
400ep	–	***95.08*** ± 0.34	*92.10* ± 1.00	–	***0.981*** ± 0.044	*0.577* ± 0.074
Bifacing 100ep	98.93 ± 0.10	94.93 ± 0.32	***93.52*** ± 0.69	0.369 ± 0.034	1.036 ± 0.047	***0.513*** ± 0.032
400ep	99.02 ± 0.09	***94.97*** ± 0.33	*93.18* ± 0.92	0.329 ± 0.028	1.022 ± 0.047	*0.520* ± 0.032
700ep	***99.15*** ± 0.08	94.91 ± 0.35	*92.97* ± 1.00	***0.325*** ± 0.031	***1.005*** ± 0.045	*0.532* ± 0.031

Bold: best training duration. Italic: no significant difference from best (two-sided Wilcoxon signed-rank test, p>0.05).

TABLE 5 | Comparison of agreement between several annotation sources.

	IoU (%; larger is better)				∆ Entrained-air (m)	
Annotator	**Overall**	**Entrained-air**	**Bad data period**	**Patch**	**MAE**	**RMSE**
Human expert (LPM)	*90.7* ± 1.2	*92.4* ± 0.9	***97.8*** ± 25.8	0.29 ± 0.10	*0.86* ± 0.10	*1.63* ± 0.17
Echoview	*88.7* ± 1.4	90.9 ± 1.1	–	–	1.05 ± 0.12	2.01 ± 0.19
Echofilter: Upfacing	*90.5* ± 2.9	***93.2*** ± 0.9	*70.7* ± 24.0	*0.32* ± 0.09	***0.76*** ± 0.05	***1.25*** ± 0.11
Echofilter: Bifacing	***91.3*** ± 1.3	*93.0* ± 1.0	*92.8* ± 25.5	***0.38*** ± 0.12	*0.78* ± 0.04	*1.27* ± 0.09

We compared several annotation methods against expert labels created by JD. The intersection-over-union (IoU) across all recordings is shown, in addition to the mean absolute error (MAE) and root-mean-square error (RMSE) for the placement of the entrained-air separation line (n = 10, ± inter-recording standard error). Note that JD used Echoview to generate seed annotations for refinement into finalized annotations. Bold: model with best agreement with target (JD) annotations. Italic: no significant difference from best (two-sided Wilcoxon signed-rank test, p>0.05).

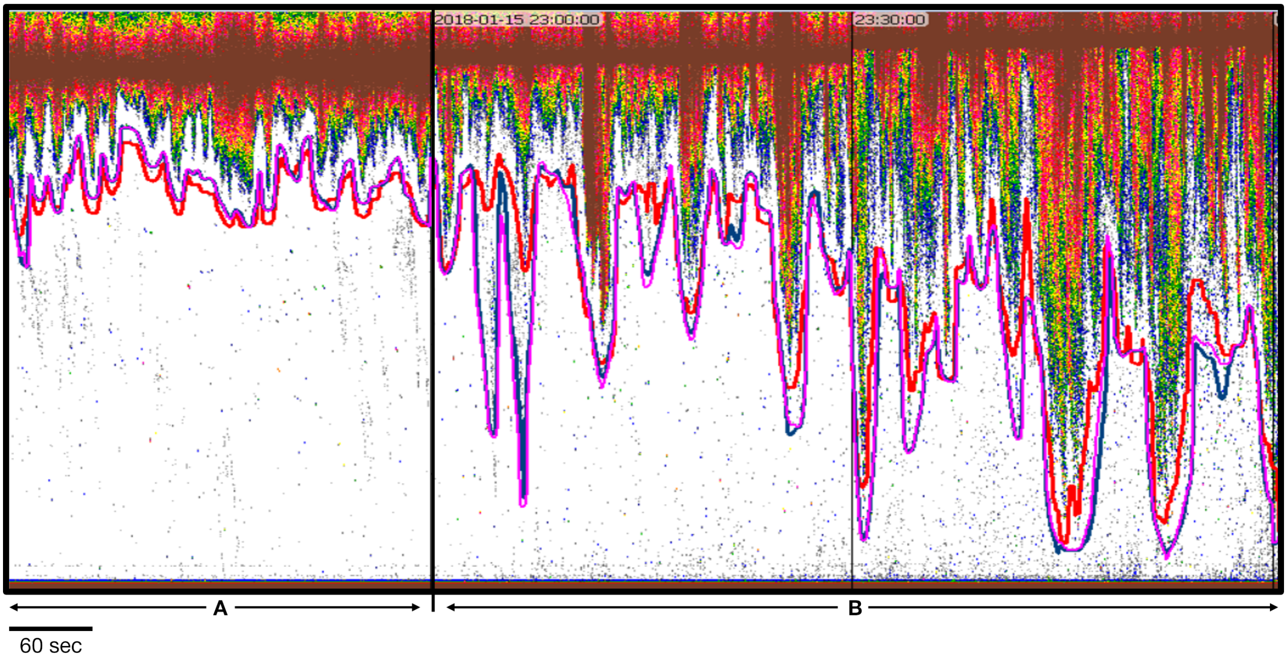

FIGURE 4 | Entrained-air lines as defined by Echofilter (Upfacing@100ep: pink; Bifacing@100ep: blue) and by Echoview (red). **(A)** A 5-minute data collection period during which entrained air does not penetrate deeply into the water. The Echoview line is further from the entrained air than the Echofilter lines, leaving greater amounts of white space and thereby unnecessarily excluding more water column from analyses. **(B)** Two 5-minute data collection periods during which the returns from entrained air are more depth dynamic. The Echofilter placement of the entrained-air lines more closely reflect the penetration of the entrained air in terms of depth and width. In the horizontal dimension, the Echofilter lines are appropriately placed further from the entrained air in the particularly steep sections. Note that Echofilter entrained-air lines as defined by each model (Bifacing@100ep and Upfacing@100ep) are essentially equivalent although not identical. Data: stationary data with echosounder in upfacing orientation, recorded for 5 minutes every half hour at the Minas Passage site.

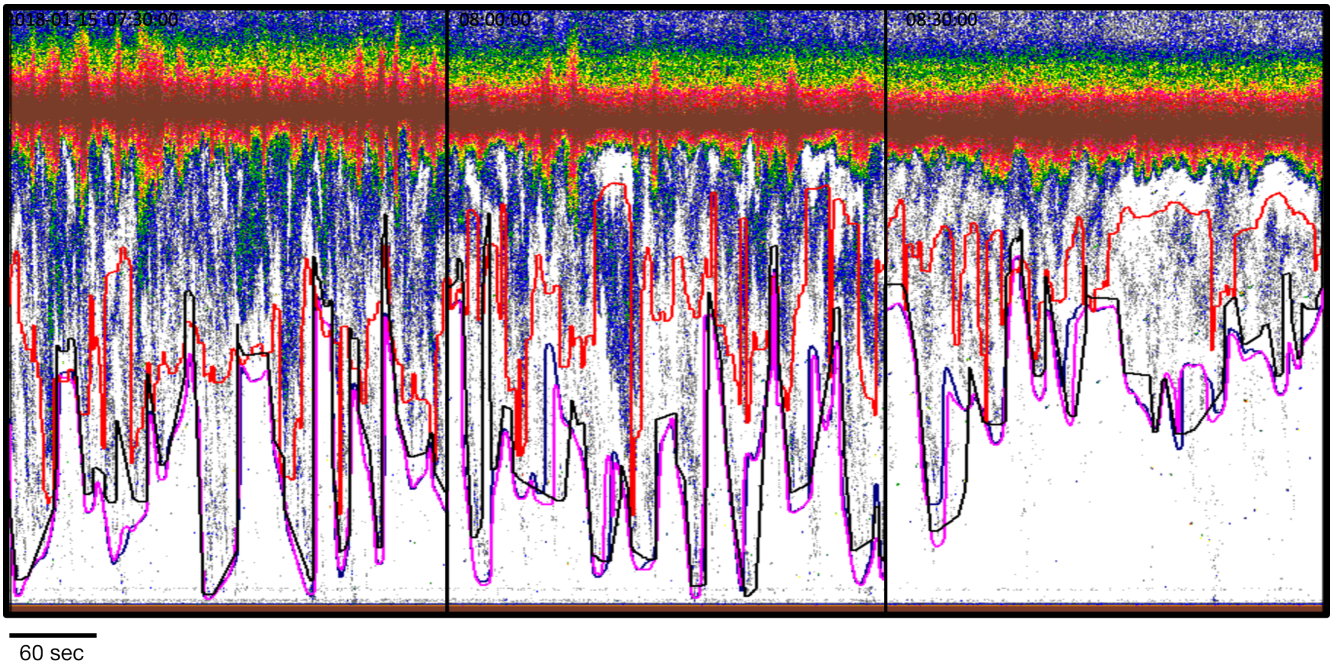

FIGURE 5 | Echogram demonstrating that the entrained-air line as calculated by the two Echofilter models (Upfacing@400ep: pink; Bifacing@700ep: blue) is a pronounced improvement over that produced by Echoview (red line), and much closer to the target line created by the analyst (black). Data: stationary data with echosounder in upfacing orientation, recorded for 5 minutes every half hour at the Minas Passage site.

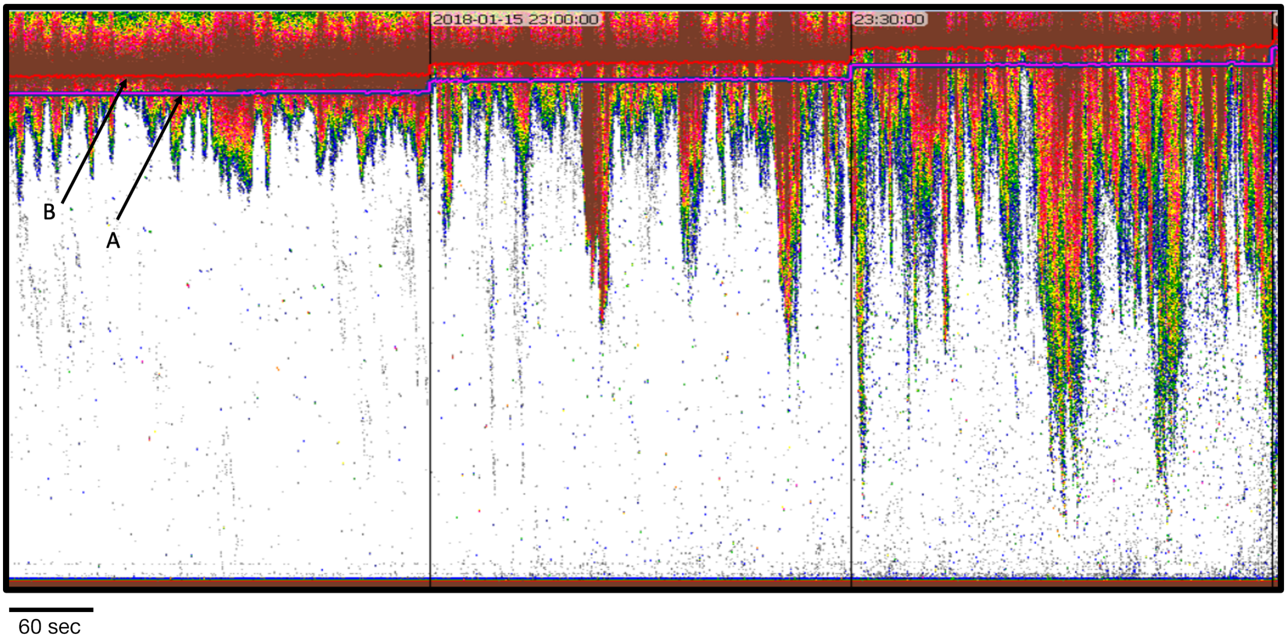

FIGURE 6 | Example showing appropriate and adequate placement of the surface line by Echofilter models: Upfacing@100ep (pink) and Bifacing@100ep (blue) indicated at **A**. The line placements include the 1 m offset required to eliminate bias from acoustic beam deadzone. For reference, the surface line, without offset, as defined by Echoview is also shown (red; line **B**). Data: stationary data with echosounder in upfacing orientation, recorded for 5 minutes every half hour at the Minas Passage site.

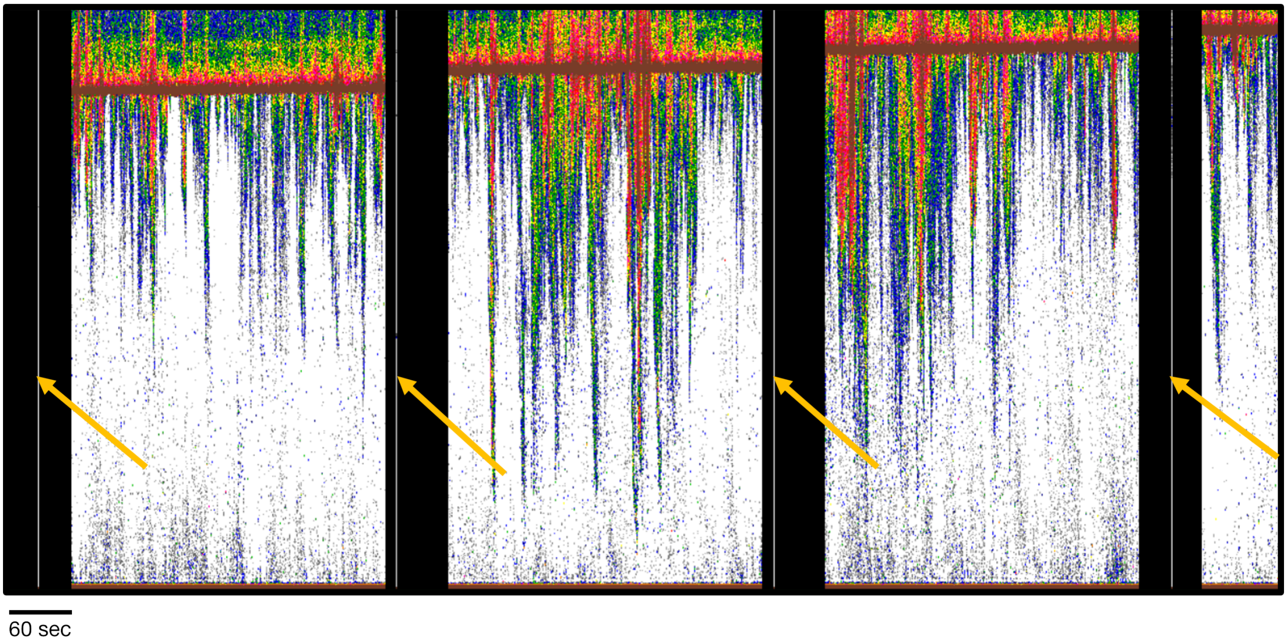

FIGURE 7 | Passive data regions (black vertical bars) as identified by Echoview algorithms. Note the white vertical lines marked by yellow arrows within the black passive data regions. The white vertical lines are single pings or a few pings misclassified by the Echoview algorithm. Data: stationary data with echosounder in upfacing orientation, active data recorded for 5 minutes and passive data for one minute every half hour at the Minas Passage site.

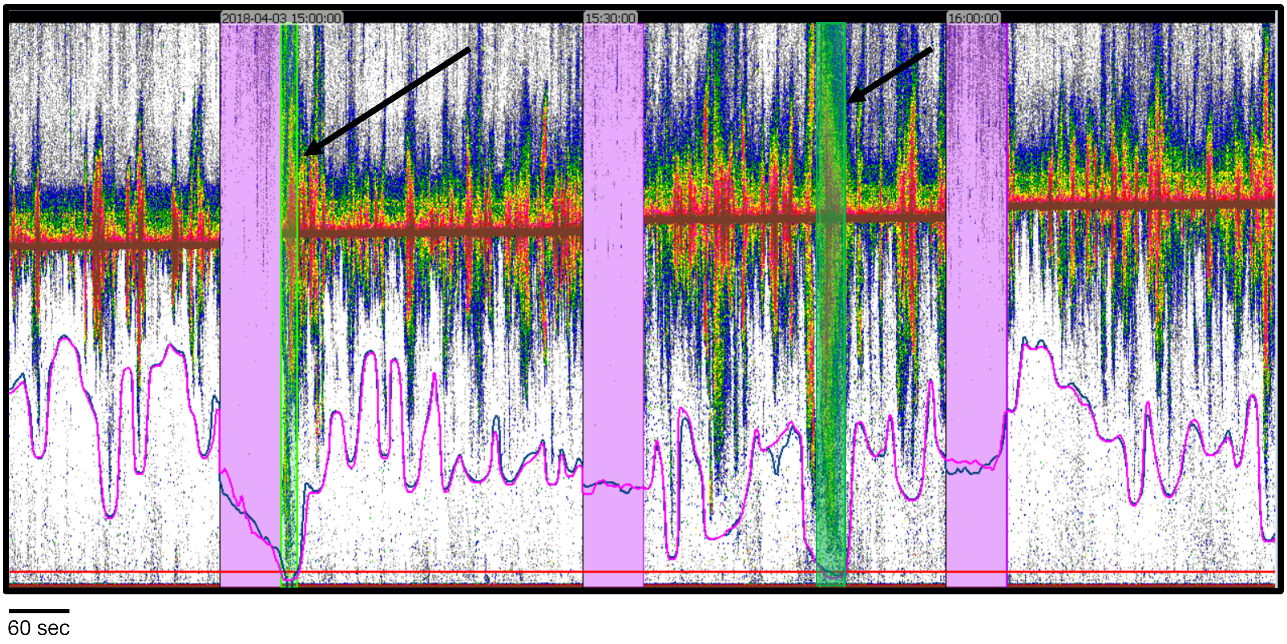

FIGURE 8 | Example of passive data regions (pink bars) and bad data periods (green bars) defined by Echofilter. Echofilter successfully identified the contiguous pings of passive data collection. Bad data periods designated by Echofilter were pings where the entrained-air line (pink or blue lines in this example) intersects with the bottom analytical line (in this case the line designating the outer boundary of the transducer nearfield). Although these cases meet the criteria, in each case the hydroacoustic analyst would designate the entire recording period as a bad data region due to the strength, penetration, and persistence of entrained-air signals. Data: stationary data with echosounder in upfacing orientation, active data recorded for 5 minutes and passive data for one minute every half hour at the Minas Passage site.

4.3 Manual Evaluation of Model Outputs

Manual investigation of the Echofilter results were carried out by JD and LPM on a Windows 10 operating system, using Echoview 10, or Echoview 11 newly released at the time of testing. The performance of Echofilter was evaluated on 24 Echoview files, selected from the test partition as described in Section 2.

During model development, a series of iterative testing and upgrades to Echofilter was undertaken. Echofilter was run on the entire set of test files, applying both models (Bifacing and Upfacing, with thresholded zoom+repeat, and logit-smoothing) to the data for comparative purposes. The results were examined for adequacy and appropriateness of the placement of lines (sea surface and entrained-air), the identification of the passive data collection periods and identification of bad data regions. Issues with the outputs were investigated in detail, and used to make changes to the model architecture design, training paradigm, or to the format of input and target data provided to the model during training. This process was iterated until any additional improvements were marginal and inconsequential.

By the end of testing and upgrades to the models, both models (Bifacing and Upfacing) produced appropriate automated initial placement of the boundary lines. Most importantly, the model placement of entrained-air boundary lines were visibly superior to the line placements as produced by the Echoview algorithms, as shown in **Figure 4**. The model results proved to be much more responsive than the Echoview algorithms to the entrained-air ambit characteristics across the varying tidal flow rates (**Figure 4**). In some cases, the automated prediction of the entrained-air line placement as produced by Echofilter were far superior to that produced by Echoview; see **Figure 5**. Note that Echofilter entrained-air lines as defined by each model (Bifacing and Upfacing) were essentially equivalent, although not identical.

As shown in **Figure 6**, the Echofilter models produced appropriate and adequate automated placement of the surface line, including a user-defined offset; in this case 1 m. Likewise, the Echofilter models produced appropriate and adequate identification of the passive data regions that will be excluded from biological analyses. We found the Echofilter passive data region identification was superior to the Echoview algorithms implemented to automate the identification of passive data regions. The Echoview algorithms would, not uncommonly, exclude a ping or few pings from within the passive data region, thereby inappropriately designating those pings for inclusion in biological analyses, as shown in **Figure 7**. No such occurrences were noted in the Echofilter results (e.g. **Figure 8**).

In addition to the passive data regions, there are two additional types of bad data regions that are not uncommon to echosounder data. The first type, is a contiguous time period marked to be removed from analysis. As shown in **Figure 8**, these bad data regions are identified by Echofilter when the position of the entrained-air line resolves to a position intersecting or extending below the bottom line, whether that line is the seafloor or the line designating the transducer nearfield exclusion line. In other words, when the position of the entrained-air line indicates that

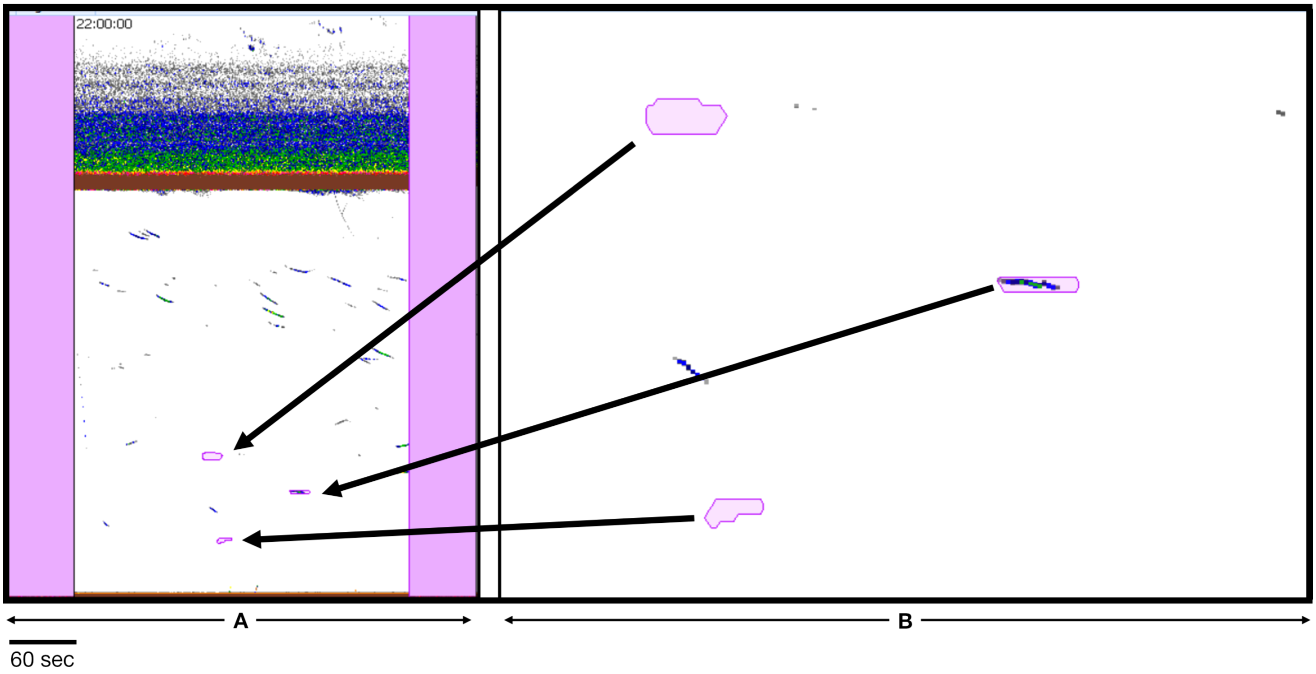

FIGURE 9 | Example of false positive "patch" bad data regions identified by Echofilter. **(A)** A 5-minute section of echogram with passive data regions (pink rectangles) on either side. **(B)** Enlargement to show the contents within each patch. Empty patches are false positive. The patch containing color samples within it would be classified as fish by the hydroacoustic analyst. It was likely identified as a bad data region by Echofilter because of its nearly horizontal position. The data on which the models were trained contain occurrences of unidentified interference which appear as horizontal lines. Those were classified as bad data regions by the analyst prior to training. Both models (Bifacing@100ep and Upfacing@100ep) designate true and false positives, but differently. Bifacing@100ep results appear to include fewer false positives. Data: stationary data with echosounder in upfacing orientation, active data recorded for 5 minutes and passive data for one minute at Minas Passage Site.

the entrained air has penetrated the entire depth of the water column. Such occurrences are not uncommon in the Minas Passage and Grand Passage datasets, sometimes occurring for just a few pings and other times the penetration occurs throughout an entire 5-minute data collection period. The single criteria of intersecting or penetrating below the bottom line is insufficient for defining all pings that should be excluded in their entirety. **Figure 8** provides an example of just such a case: less than 50% of the water column remains after the entrained-air exclusion. In that case, if the goal of the analyses is to understand metrics within the full water column, that data collection period would need to be excluded in its entirety.

The second type of bad data region, a "patch" of bad data, can be characterized as forming randomly shaped discrete patches. Within the original test segment of 24 files, only three had occurrences of the patch-type bad data region. Two additional EV files containing patch-type bad data regions were identified from the validation and training segments for manual inspection of the patch-type results only. Both Echofilter models performed poorly, generating false positives as illustrated in **Figure 9**.

TABLE 6 | Results from the time-to-edit experiment.

		Edit order		Edit time (MM:SS)		
Tide	**Phase**	**Echoview**	**Echofilter**	**Echoview**	**Echofilter**	**Reduction**
Spring	Flood	2	6	8:06	**4:04**	50%
Spring	Ebb	3	4	8:00	**4:04**	49%
Neap	Flood	8	7	8:18	**4:30**	46%
Neap	Ebb	1	5	7:08	**3:57**	45%
Bad file		9	10	4:28	**1:51**	59%
Overall Mean				7:12	***3.42***	49%

A hydroacoustic analyst used the entrained-air lines produced by either Echoview or Echofilter to seed their annotations. We compare the amount of time needed to convert the seed lines into "correct" annotation lines. Bold: best model (shortest duration).

4.4 Time-Savings Analysis

We sought to evaluate the amount of time-savings that the Echofilter model would offer, relative to the existing workflow using Echoview algorithms. Five of the Echoview files from the MP:sta↑ test partition were selected for a time test. The files were selected to represent each tide and phase combination: flooding spring tide, ebbing spring tide, flooding neap tide, and ebbing neap tide, plus one file with especially noisy data for which neither Echoview or Echofilter would likely render a well-placed entrained-air line. Annotations were initialized twice: once using the preexisting workflow utilizing Echoview algorithms, and once using Echofilter with the Upfacing@100ep model, with logit-smoothing enabled. The initial entrained-air line in each of the ten files was audited and edited by the hydroacoustic analyst

(JD), while recording the amount of time taken to do so. We randomized the order in which tasks (file and seed annotation source) were completed, except the especially noisy "bad file" which was evaluated later.

Our results, shown in **Table 6**, demonstrate that using the annotations generated by Echofilter results in less time taken for the human annotator to complete their task. For typical data files, the time taken to finalize annotations was consistently 45%–50% shorter when using annotations produced by Echofilter as the seed instead of annotations produced by Echoview. For an especially noisy file, the reduction in time was even larger, at 59%. The reduction in time was statistically significant (p<0.001; paired Student's t-test).

5 DISCUSSION

5.1 Impact of Echofilter Model

We have described the implementation of a deep learning model, Echofilter, which can be used to generate annotations to segment entrained air appearing in hydroacoustic recordings at tidal energy sites. Our goal was to produce an automated, model-based approach to the placement of a line appropriately defining the boundary between that portion of the water column contaminated by acoustic returns from entrained air, and that portion of the water column appropriate for biological analyses. This was motivated by the need for reliable, timely analyses and subsequent reporting to assist regulators, developers, and stakeholders in understanding the risks to fish imposed by the deployment of tidal energy devices into marine ecosystems.

We found the deep learning models we implemented produced significantly and appreciably better placement of the entrained-air line than the Echoview algorithms. For mobile, downward-facing recordings, the average error was 0.33 m, less than a third of Echoview's 1.2 m average error. For stationary, upward-facing recordings, the average error was 0.5 m to 1.0 m depending on dataset, consistently less than half the error seen with Echoview algorithms (1.2 m to 2.2 m). Furthermore, the surface, seafloor, and passive region placement were also superior to those produced using Echoview. The model's overall annotations had a high level of agreement with the human segmentation, with an intersection-over-union score of 99% for mobile downfacing recordings and 92% to 95% for stationary upfacing recordings. As such, Echofilter provides a complete automated line placement and passive data identification methodology.

The most challenging segmentation line to place correctly is the entrained-air line, which currently can require time-consuming manual placement due to the lack of a well-placed automated solution. We found that the increase in accuracy of the automated placement of the entrained-air line provided by Echofilter corresponded to a 50% reduction in the time required for a hydroacoustician to audit and correct the line placement. Our quantitative analysis has shown that the Echofilter models produce lines which are closer (in distance) to the line placement defined by the human expert. Additionally, we note that the ML models are more sensitive to the fine-scale nuances in the boundary position of the entrained air; when the model places the line incorrectly, the errors tend follow the correct shape of the entrained air but are offset by some amount, and hence require only a simple, coarse edit to shift the line in some region to the correct offset. In contrast, when the Echoview algorithm is incorrect, the shape is incorrect and corrections to the line involve time-consuming fine-scale edits instead. Since coarse-scale edits are less cognitively taxing and far fewer edits are required, far less analyst fatigue is invoked during manual corrections of the model-placed entrained-air line, thereby allowing the analyst to bring the full-force of their intellect, training, and analytical skills to modifying placement of the line segments as necessary. Additionally, the reduction in the number of fine-scale edits provides the opportunity for an increase in the standardization and repeatability of line placement, within an analyst's work and among analysts.

Machine learning applied to the hydroacoustic data by which we quantify fish distribution and abundance has garnered improvements to the work flow and increased the efficiency of the work by 50%, improvements that haven't been achieved any other way. The machine learning contribution to assessing the ecological impacts of introducing marine renewable energy devices into the marine habitat is the improved analytical consistency and substantial improvements in the timeliness of analyses and subsequent reporting.

5.2 Limitations Associated With Echofilter

We developed Echofilter with the goal of increasing the efficiency and standardization of the post-processing of hydroacoustic data collected in dynamic marine environments such as tidal channels. The model was thoroughly evaluated on data recorded from upward-facing stationary echosounders at two tidal energy demonstration sites in the Bay of Fundy. The models have not been evaluated on data collected in other regions, with other instrumentation, or in other deployment configurations. Consequently, the performance of Echofilter on data collected under conditions that differ substantially from those used for model development may be heavily impacted and require some level of re-training to ensure accurate results, which is a non-trivial procedure.

In addition to the entrained-air boundary line, Echofilter predicts the depths of the surface (for upfacing recordings) and the seafloor (downfacing). Our performance metrics indicate that these lines are all placed accurately, however we have not thoroughly inspected the model's output on downfacing recordings and can not confirm the integrity of the seafloor line.

In addition to the lines, our model attempts to predict regions which should be excluded from biological analyses. However, it was not possible for the model to learn these annotations with sufficient accuracy to be usable for downstream tasks. Consequently, it is not possible to automate away a need for manual inspection of the data. A hydroacoustician must always inspect the recordings themselves in order to annotate regions to exclude from analysis, and adjust lines as necessary.

5.3 Accessing Echofilter

To ensure the broader community can utilize our model described in this paper, we have released the final implementation, Echofilter, under the AGPLv3 license. Python source code and a stand-alone Windows executable are available at https://github.com/DeepSenseCA/echofilter. Additionally, the command line interface (CLI) and application programming interface (API) documentation is available at https://DeepSenseCA.github.io/echofilter/.

We hope this tool will prove useful to tidal energy researchers, and the wider hydroacoustic community.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation. Requests to access these datasets should be directed to FORCE, info@fundyforce.ca.

AUTHOR CONTRIBUTIONS

Manuscript written by SL and LM. Data conversion was performed by JN and SL. Model architecture design and training was performed by SL. Interface and API development by SL. Model evaluation was performed by LM and JD. Identification of features of specific value to the hydroacoustics community was performed by LM. Project oversight by DH, CW, and SO. All authors contributed to the article and approved the submitted version.

FUNDING

SL was funded by Mitacs award IT16140. LPM was supported by the Offshore Energy Research Association of Nova Scotia (OERA) under the Pathway Program; funding provided by Natural Resources Canada, grant number ERPP-RA-05.

ACKNOWLEDGMENTS

This work was supported in part by Mitacs through the Mitacs Accelerate program, and by Offshore Energy Research Association (OERA; https://oera.ca/). Computations were performed on the DeepSense (https://deepsense.ca/) high-performance computing platform. DeepSense is funded by ACOA, the Province of Nova Scotia, the Centre for Ocean Ventures and Entrepreneurship (COVE), IBM Canada Ltd., and the Ocean Frontier Institute (OFI). We thank Echoview Software Pty Ltd. and Haley Viehman, Ph.D., for providing support and technical advice for this project. We thank Paul C. Hines of Hines Ocean S&T for his helpful discussions during preparation of the manuscript. Additionally, we thank Tyler Boucher, Aurélie Daroux, Mili Sanchez, and Haley Viehman, for (in addition to JD and LM) creating the expert annotations for the data used in this study. This work has previously appeared as an arXiv preprint (Lowe et al., 2022), available at doi: 10.48550/arxiv.2202.09648.

REFERENCES

Bengio, Y., Lecun, Y. and Hinton, G. (2021). Deep Learning for AI. *Commun. ACM* 64, 58–65. doi: 10.1145/3448250

Benoit-Bird, K. J. and Lawson, G. L. (2016). Ecological Insights From Pelagic Habitats Acquired Using Active Acoustics. *Annu. Rev. Mar. Sci.* 8, 463–490. doi: 10.1146/annurev-marine-122414-034001

Blaber, S. J. M., Cyrus, D. P., Albaret, J.-J., Ching, C. V., Day, J. W., Elliott, M., et al. (2000). Effects of Fishing on the Structure and Functioning of Estuarine and Nearshore Ecosystems. *ICES J. Mar. Sci.* 57, 590–602. doi: 10.1006/jmsc.2000.0723

Cada, G., Ahlgrimm, J., Bahleda, M., Bigford, T., Stavrakas, S. D., Hall, D., et al. (2007). Potential Impacts of Hydrokinetic and Wave Energy Conversion Technologies on Aquatic Environments. *Fisheries* 32, 174–181. doi: 10.1577/1548-8446(2007)32[174:PIOHAW]2.0.CO;2

Copping, A. E., Hemery, L. G., Overhus, D. M., Garavelli, L., Freeman, M. C., Whiting, J. M., et al. (2020). Potential Environmental Effects of Marine Renewable Energy Development—the State of the Science. *J. Mar. Sci. Eng.* 8. doi: 10.3390/jmse8110879

Cornett, A., Toupin, M. and Nistor, I. (2015). "Appraisal of the IEC Technical Specification for Tidal Energy Resource Assessment at Minas Passage, Bay of Fundy, Canada," in *Proc. 2015 European Wave and Tidal Energy Conference (EWTEC)* (Nantes, France).

DFO (2008). *Potential Impacts of, and Mitigation Strategies for, Small-Scale Tidal Generation Projects on Coastal Marine Ecosystems in the Bay of Fundy* (Tech. rep., Fisheries and Oceans Canada (DFO) Canadian Science Advisory Secretariat). Science Response 2008/013.

DFO (2018). *Delineating Important Ecological Features of the Evangeline-Cape Blomidon-Minas Basin Ecologically and Biologically Significant Area (EBSA)* (Tech. rep., Fisheries and Oceans Canada (DFO) Canadian Science Advisory Secretariat). Science Response 2018/005.

Fernandes, P., Gerlotto, F., Holliday, D., Nakken, O. and Simmonds, E. (2002). Acoustic Applications in Fisheries Science: The ICES Contribution. *ICES Mar. Sci. Symp.* 215, 483–492. doi: 10.17895/ices.pub.8889

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning* (MIT Press). Available at: http://deeplearningbook.org

Guerra, M., Hay, A. E., Karsten, R., Trowse, G. and Cheel, R. A. (2021). Turbulent Flow Mapping in a High-Flow Tidal Channel Using Mobile Acoustic Doppler Current Profilers. *Renewable Energy* 177, 759–772. doi: 10.1016/j.renene.2021.05.133

He, K., Zhang, X., Ren, S. and Sun, J. (2016). "Deep Residual Learning for Image Recognition," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (*New York, NY, USA: IEEE*)*, pp. 770–778. doi: 10.1109/CVPR.2016.90

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861. doi: 10.48550/arxiv.1704.04861.

Hu, J., Shen, L., Albanie, S., Sun, G. and Wu, E. (2019). Squeeze-And-Excitation Networks. arXiv preprint arXiv:1709.01507. doi: 10.48550/arxiv.1709.01507.

IPCC (2021). "Summary for Policymakers," in *Climate Change 2021: The Physical Science Basis. Contributions of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Eds. Masson-Delmotte, V., Zhai, P., Pirani, A., Connors, S. L., Péan, C., Berger, S., Caud, N., Chen, Y., Goldfarb, L., Gomis, M. I., Huang, M., Leitzell, K., Lonnoy, E., Matthews, J. B. R., Maycock, T. K., Waterfield, T., Yelekçi, O., Yu, R. and Zhou, B. (Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press), 3–32.

IRENA (2020). *Innovation Outlook: Ocean Energy Technologies* Abu Dhabi: Tech. rep., International Renewable Energy Agency (IRENA).

Jaccard, P. (1912). The Distribution of the Flora in the Alpine Zone. *New Phytol.* 11, 37–50. doi: 10.1111/j.1469-8137.1912.tb05611.x

Johannesson, K. A. and Mitson, R. B. (1983). *Fisheries Acoustics: A Practical Manual for Aquatic Biomass Estimation,* vol. 240 *of FAO Fisheries Technical Paper* (Rome, Italy: Food and Agriculture Organization of the United Nations).

Karpathy, A. (2014). What I Learned From Competing Against a ConvNet on ImageNet. In: *Andrej Karpathy Blog*. Available at: http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/ (Accessed 2022-05-01).

Karsten, R., Greenberg, D., Tarbotton, M., Culina, J., Swan, A., O'Flaherty-Sproul, M., et al. (2011). *Assessment of the Potential of Tidal Power From Minas Passage and Minas Basin* (Acadia University). Tech. Rep. 300-170-09-11.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). "ImageNet Classification With Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 25. Eds. Pereira, F., Burges, C., Bottou, L. and Weinberger, K. (Redhook, NY, USA: Curran Associates, Inc).

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep Learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., et al. (2020). "On the Variance of the Adaptive Learning Rate and Beyond," In *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*, doi: 10.48550/arXiv.1908.03265

Lowe, S. C., McGarry, L. P., Douglas, J., Newport, J., Oore, S., Whidden, C., et al. (2022). Echofilter: A Deep Learning Segmentation Model Improves the Automation, Standardization, and Timeliness for Post-Processing Echosounder Data in Tidal Energy Streams. arXiv preprint arXiv:2202.09648. doi: 10.48550/arxiv.2202.09648.

Lowe, S. C., Trappenberg, T. and Oore, S. (2021). LogAvgExp Provides a Principled and Performant Global Pooling Operator. arXiv preprint arXiv:2111.01742. doi: 10.48550/arxiv.2111.01742.

Melvin, G. D. and Cochrane, N. A. (2012). *A Preliminary Investigation of Fish Distributions Near an In-Stream Tidal Turbine in Minas Passage, Bay of Fundy* (Tech. rep., Canadian Technical Report of Fisheries and Aquatic Sciences), Tech. rep. 3006.

Melvin, G. D. and Cochrane, N. A. (2015). Multibeam Acoustic Detection of Fish and Water Column Targets at High-Flow Sites. *Estuaries Coasts* 38, 227–240. doi: 10.1007/s12237-014-9828-z

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2022). Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 44, 3523–3542. doi: 10.1109/TPAMI.2021.3059968

Perez, L., Cossu, R., Grinham, A. and Penesis, I. (2021). Seasonality of Turbulence Characteristics and Wave-Current Interaction in Two Prospective Tidal Energy Sites. *Renewable Energy* 178, 1322–1336. doi: 10.1016/j.renene.2021.06.116

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). "You Only Look Once: Unified, Real-Time Object Detection," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (*New York, NY, USA: IEEE*)*, 779–788. doi: 10.1109/CVPR.2016.91

Roberts, A., Thomas, B., Sewell, P., Khan, Z., Balmain, S. and Gillman, J. (2016). Current Tidal Power Technologies and Their Suitability for Applications in Coastal and Marine Areas. *J. Ocean Eng. Mar. Energy* 2, 227–245. doi: 10.1007/s40722-016-0044-8

Ronneberger, O., Fischer, P. and Brox, T. (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Eds. Navab, N., Hornegger, J., Wells, W. M. and Frangi, A. F. (Cham: Springer International Publishing), 234–241. doi: 10.1007/978-3-319-24574-4_28

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vision* 115, 211–252. doi: 10.1007/s11263-015-0816-y

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D. and Lillicrap, T. (2016). One-Shot Learning With Memory-Augmented Neural Networks. arXiv preprint arXiv:1605.06065. doi: 10.48550/arxiv.1605.06065.

Schmidhuber, J. (2015). Deep Learning. *Scholarpedia* 10, 32832. doi: 10.4249/scholarpedia.32832. Revision 184887.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., et al. (2017). Mastering the Game of Go Without Human Knowledge. *Nature* 550, 354–359. doi: 10.1038/nature24270

Simmonds, J. and MacLennan, D. (2005). *Fisheries Acoustics Theory and Practice*. 2nd edn (Oxford, UK: Blackwell Publishing).

Smith, L. N. (2015). No More Pesky Learning Rate Guessing Games. arXiv preprint arXiv:1506.01186. doi: 10.48550/arxiv.1506.01186.

Smith, L. N. (2018). A Disciplined Approach to Neural Network Hyper-Parameters: Part 1 – Learning Rate, Batch Size, Momentum, and Weight Decay. arXiv preprint arXiv:1803.09820. doi: 10.48550/arxiv.1803.09820.

Smith, L. N. and Topin, N. (2017). Super-Convergence: Very Fast Training of Residual Networks Using Large Learning Rates. arXiv preprint arXiv:1708.07120. doi: 10.48550/arxiv.1708.07120.

Tan, M. and Le, Q. (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, vol. 97. 6105–6114. doi: 10.48550/arxiv.1905.11946

Tong, Q., Liang, G. and Bi, J. (2022). Calibrating the Adaptive Learning Rate to Improve Convergence of ADAM. *Neurocomputing* 481, 333–356. doi: 10.1016/j.neucom.2022.01.014

Tsitrin, E., Sanderson, B. G., McLean, M. F., Gibson, A. J. F., Hardie, D. C. and Stokesbury, M. J. W. (2022). Migration and Apparent Survival of Post-Spawning Alewife (*Alosa pseudoharengus*) in Minas Basin, Bay of Fundy. *Anim. Biotelemetry* 10:11. doi: 10.1186/s40317-022-00277-z

Williamson, B. J., Fraser, S., Blondel, P., Bell, P. S., Waggitt, J. J. and Scott, B. E. (2017). Multisensor Acoustic Tracking of Fish and Seabird Behavior Around Tidal Turbine Structures in Scotland. *IEEE J. Oceanic Eng.* 42, 948–965. doi: 10.1109/JOE.2016.2637179

Wolf, J., Dominicis, M. D., Lewis, M., Neill, S. P., O'Hara Murray, R., Scott, B., et al. (2022). "9.04 - Environmental Issues for Offshore Renewable Energy," in *Comprehensive Renewable Energy*, 2nd edn. Ed. Letcher, T. M. (Oxford: Elsevier), 25–59. doi: 10.1016/B978-0-12-819727-1.00036-4

Wright, L. (2019). "Ranger - a Synergistic Optimizer," in *GitHub Repository*. Available at: https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer. Revision @8d636a5.

Yong, H., Huang, J., Hua, X. and Zhang, L. (2020). "Gradient Centralization: A New Optimization Technique for Deep Neural Networks," in *Computer Vision – ECCV 2020*. Eds. Vedaldi, A., Bischof, H., Brox, T. and Frahm, J.-M. (Cham: Springer International Publishing), 635–652. doi: 10.48550/arxiv.2004.01461

Zhang, M., Lucas, J., Ba, J. and Hinton, G. E. (2019). "Lookahead Optimizer: K Steps Forward, 1 Step Back," in *Advances in Neural Information Processing Systems*, vol. vol. 32 . Eds. Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. and Garnett, R. (Redhook, NY, USA: Curran Associates, Inc). doi: 10.48550/arxiv.1907.08610

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.