Machine learning-based region of interest detection in airborne lidar fisheries surveys

Trevor C. Vannoy,[a] **Jackson Belford**,[a] **Joseph N. Aist**,[a] **Kyle R. Rust**,[a] **Michael R. Roddewig**,[a,b] **James H. Churnside**,[c,d] **Joseph A. Shaw**,[a,b] **and Bradley M. Whitaker**[a,b,*]

[a]Montana State University, Electrical and Computer Engineering, Bozeman, Montana, United States

[b]Montana State University, Optical Technology Center, Bozeman, Montana, United States

[c]National Oceanic and Atmospheric Administration, Chemical Sciences Laboratory, Boulder, Colorado, United States

[d]University of Colorado, Cooperative Research in the Environmental Sciences, Boulder, Colorado, United States

Abstract. Airborne lidar data for fishery surveys often do not contain physics-based features that can be used to identify fish; consequently, the fish must be manually identified, which is a time-consuming process. To reduce the time required to identify fish, supervised machine learning was successfully applied to lidar data from fishery surveys to automate the process of identifying regions with a high probability of containing fish. Using data from Yellowstone Lake and the Gulf of Mexico, multiple experiments were run to simulate real-world scenarios. Although the human cannot be fully removed from the loop, the amount of data that would require manual inspection was reduced by 61.14% and 26.8% in the Yellowstone Lake and Gulf of Mexico datasets, respectively. [DOI: 10.1117/1.JRS.15.038503]

Keywords: lidar; machine learning; fisheries; water.

Paper 210023 received Jan. 13, 2021; accepted for publication Jul. 2, 2021; published online Jul. 19, 2021.

1 Introduction

Setting annual catch limits of fish is difficult because individual stocks vary widely as a result of fishing pressure and environmental factors such as the availability of food, predation levels, and larval survival. Knowledge of these stocks is critical to setting appropriate catch limits, but this knowledge is often imprecise. For example, a 2003 paper in the journal *Nature* claimed that "large predatory fish biomass today is only about 10% of pre-industrial levels."[1] The conclusion was based on a Catch Per Unit Effort analysis, in which commercial fish landings and the time spent fishing are combined to estimate fish biomass. The analysis was controversial[2,3] and speaks to the desirability of an estimate of biomass that does not depend on the fishery in question.

The two main fishery-independent survey types are controlled catches and acoustics. In controlled catches, it is the effort that is controlled. The same gear is used the same way in the same place at the same time of year each year to build up a time series of catches that can be related to increases or decreases in the fish stock. The objective is to statistically sample the entire habitat, unlike the commercial fleet that goes where the fish are in greatest abundance. With acoustics, the ship can continuously sample along the track to obtain more data, increasing the statistical significance. Calibration factors allow the conversion of acoustic backscatter to biomass. Generally, acoustic surveys also employ selective fishing to verify species identification and obtain information about the size, health, and reproductive status of the fish. These, and other

*Address all correspondence to Bradley M. Whitaker, bradley.whitaker1@montana.edu

1931-3195/2021/$28.00

less common techniques, are all ship-based and are limited to the speed of a surface vessel, and these long surveys can be very expensive.

1.1 *Airborne Lidar for Fishery Surveys*

Increasing the speed at which surveys can be conducted requires aerial surveys,[4,5] which have mostly been done visually or with camera systems. However, fish can be detected much deeper in the water column by lidar,[6] and airborne lidar has been explored for fishery surveys in oceans[7–10] and lakes.[11,12] Related applications of airborne lidar include mapping marine debris,[13,14] plankton layers,[15–17] bubbles,[18,19] internal waves[14] in the ocean, underwater thermal vents,[20] and water turbidity.[21] Initial research has started assessing the potential of spaceborne lidars for oceanic profiling of plankton and other biological and particulate matter.[22–30]

All airborne lidar applications produce large amounts of data. Unlike lidar bathymetry[31–33] and ocean lidar profiling,[21,34] most lidar data for fishery surveys do not contain physics-based features that can be used to easily identify fish or other objects of interest. As a result, identifying and marking the positions of fish in lidar images can take 10 to 20 min per hour of lidar data. Data with no fish are faster to process than data with large numbers of fish, but that is about average. As a concrete example, the Gulf of Mexico survey[10] analyzed in this work contains ~55 h of lidar data; visually inspecting this data took about 14 h. Consequently, there is a need for automated methods that can identify regions of fish and reduce the amount of data that requires manual inspection.

Previous efforts to automate lidar processing for fisheries have used a combination of spatial filtering and thresholding to locate schools of mackerel[35] and swarms of zooplankton.[36] These have been limited by the difficulty in optimizing filter parameters and threshold levels for the situation of interest. The situation of interest includes the return strength and spatial variability of the background water return, as well as the strength and spatial characteristics of the target species. Due to the limitations of previous efforts, visual inspection is still the state of the art. In an effort to develop an automated processing technique, this paper investigates the feasibility of using supervised machine learning to identify regions containing fish.

1.2 *Supervised Learning on Imbalanced Data*

Supervised learning is a type of machine learning that learns how to map input data to output labels by training on example input-output pairs. As a result, supervised algorithms can be trained to minimize misclassification rates. Unsupervised algorithms, in contrast, are designed to find associations between input data and possible output groups. Without a knowledge of class labels, the training procedure cannot be directly optimized for prediction accuracy.[37] For this purpose, our work focuses on supervised learning techniques. The wide variety of available supervised learning algorithms makes it infeasible to perform an exhaustive search. Therefore, we focus on comparing results obtained from a few well-known algorithms—support vector machines (SVM),[38] decision trees,[39] linear discriminant analysis (LDA),[40] and neural networks[41]—and RUSBoost,[42] an algorithm designed specifically for imbalanced data.

We acknowledge that the ability of supervised classifiers to produce more accurate results comes at a cost: they require labeled data for the training process. Because of this, it is impossible to develop a completely autonomous detection system using only supervised classification. Prior to completing the training process, some data need to be manually inspected and labeled, so the classifiers have a ground truth to learn from. For example, during a multi-day lidar campaign, a human may manually inspect the lidar results from the first day. The labels from the first day can then be used to create a classifier for automated detection during subsequent days. It is also important to note that, since ground truth labels are prone to human error, the reported classification results may be imperfect. However, this does not always have negative consequences. For example, classifiers sometimes find a fish that a human missed. Such situations negatively affect performance metrics, but ultimately result in more fish detections.

When training on data with a large class imbalance, such as lidar data from fishery surveys in which there are very few fish examples, many traditional machine learning techniques will fail to

identify the minority class(es).[43–45] Consider one formulation of the optimization problem that must be solved for a non-separable SVM:[37]

$$\min \|\beta\| \text{ subject to} \begin{cases} y_i(x_i^T\beta + \beta_0) \geq 1 - \zeta_i \ \forall \ i \\ \zeta_i \geq 0, \ \ \sum \zeta_i \leq \text{constant} \end{cases},$$

where β and β_0 are the normal vector and y-intercept of the optimized hyperplane (decision boundary), respectively, x_i is measurement vector i with associated binary label $y_i = \pm 1$, and ζ_i is a slack variable that allows for individual points to be on the wrong side of the hyperplane. It is worth noting that, in this formulation, each input vector exerts the same amount of force on the hyperplane. This can cause the majority class to overwhelm the minority class, resulting in a classifier that skews heavily toward classifying new points as part of the majority class.

Several techniques have been developed to improve results in situations in which the classes are imbalanced.[46] One common method is to weight the classes according to their distributions by imposing a higher cost for misclassifying vectors from the minority class. Two other methods are sampling and boosting.[47] RUSBoost,[42] which is used in this paper, combines random undersampling and boosting to improve classification performance on imbalanced data. The primary methods used to deal with class imbalance in this study are undersampling and class weighting, as they can both be applied to existing classification algorithms without modification.

2 Lidar Systems

Currently, there are no commercial lidar systems available for this application. Fish have been detected by commercial bathymetric lidars but only as an artifact that must be eliminated during processing.[48] Terrestrial lidars not designed for bathymetry typically use laser wavelengths that do not penetrate significantly into water. However, the technology for a lidar to detect subsurface fish and plankton layers is much simpler than a bathymetric lidar, and several have been constructed in the United States,[8,11,49] Russia,[50–52] and China.[53,54] Two of these are considered in this paper.

2.1 *NOAA Lidar*

The NOAA lidar is a down-looking, aircraft-based instrument. The light source for the transmitter of the lidar is a frequency-doubled, Q-switched Nd:YAG laser. The laser itself emits linearly polarized, 532-nm light in 12-ns pulses at a repetition rate of 30 Hz. The laser light is directed through a polarization beam splitting cube (103 polarization extinction ratio), beam-steering mirrors, and a negative lens to ensure that the light reaching the water surface is within the American National Standards Institute standards for exposure to laser light.[55] The pulse energy at the output of the transmission optics is 100 mJ. On the receiver side, the lidar employs two channels for measuring the return light: one for the component of the return with the same polarization as the transmitter (co-polarized return) and the other for its orthogonal polarization (cross-polarized return). For each channel, the received light is collected by a telescope, filtered by a 1-nm bandwidth interference filter, and detected by a photomultiplier tube (PMT). The 17-mrad field of view of each telescope is set to match the transmitter beam divergence. A plastic-film linear polarizer is installed at the front of each telescope to select the appropriate linear polarization state. The photomultiplier signal is amplified through a logarithmic amplifier and digitized at 1 GHz. For this study, the lidar was mounted in the floor of a small twin-engine aircraft and flown at an altitude of ~300 m over the ocean surface. The lidar system was pointed ~15 deg from nadir to minimize contribution from air-water interface reflections.

2.2 *MSU Lidar*

The MSU lidar was developed in 2014 and shares many similarities with the NOAA lidar. It uses a 532-nm pulsed Nd:YAG laser with a 26.8-mJ pulse energy and 100-Hz repetition frequency. The lidar has a selectable 5 or 15 mrad FOV, which at the typical flight altitude of 300 m gives a

1.5- or 4.5-m diameter spot size, respectively. Two orthogonally polarized receiver telescopes are used to simultaneously measure the co- and cross-polarization return; PMT and synchronized 800 MSPS digitizers convert the returned optical energy to digital numbers, which are stored for later analysis.[11] This lidar is used extensively in Yellowstone National Park to map lake trout spawning locations;[12] it recorded the first known lidar detection of an underwater thermal vent[20] and characterized the lidar attenuation coefficient.[21]

3 Datasets

Two airborne lidar datsets were used in this study: one comprising lake trout in Yellowstone lake[12] and the other comprising flying fish, schools of unidentified fish, jellyfish, and plankton in the Gulf of Mexico.[10]

3.1 *Yellowstone Lake Dataset*

Discovered in 1994,[56] invasive lake trout are now the top carnivore in Yellowstone Lake, Wyoming[57,58]—threatening the ecological balance of Yellowstone National Park. The Yellowstone Lake survey[12] was conducted to determine if airborne lidar could be used to detect the invasive lake trout. This dataset was selected for this work due to the application's ecological significance and availability of the data.

The survey was conducted with daytime flights during lake trout spawning season in 2015 and 2016[12] using the airborne lidar system described in Sec. 2.2. Roddewig et al.[12] manually identified fish in the lidar data, which we use as ground truth labels. A summary of the Yellowstone Lake dataset is shown in Table 1, and a representative example of lidar data is shown in Fig. 1.

Table 1 Summary of data collected at Yellowstone Lake,[11] showing the number of instances (lidar shots), dimensions (samples per shot), and instances containing fish. The percentage of fish instances are shown in parentheses. A single fish often shows up in multiple adjacent instances.

	Instances	Dimensions	Fish instances
2015 Flight	108778	2048	401 (0.369%)
2016 Flight	192201	2048	227 (0.118%)

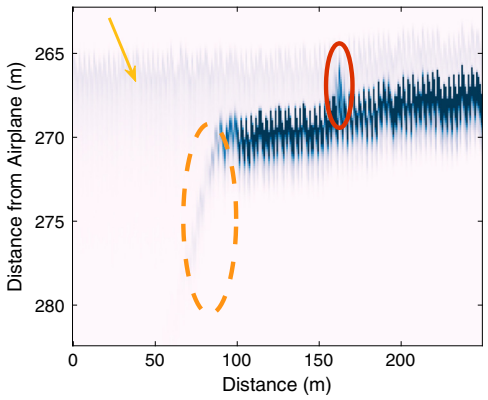

Fig. 1 Example cross-polarized lidar data from Yellowstone Lake. Darker values indicate higher reflected radiance; radiance values were compressed to increase contrast. The light patch indicated by the yellow arrow is the surface of the water. The start of an underwater shelf is indicated by the orange dashed ellipse. The red ellipse highlights a typical fish hit, which appears as a vertical spike above the shelf.

Table 2 Summary of data collected over the Gulf of Mexico,[10] showing the number of instances (lidar shots), dimensions (samples per shot), and instances containing fish, schools of fish, jellyfish, and plankton layers, respectively. The percentages of each instance type are shown in parentheses. Objects of interest often occur across multiple adjacent instances. Some instances contain multiple types of objects of interest. Day 11 is excluded due to the lack of ground truth labels.

	Instances	Dimensions	Fish instances	Fish school Instances	Jellyfish instances	Plankton layer instances	All fish instances
Day 1	747284	1000	7232 (0.968%)	217 (0.0290%)	0	106854 (14.3%)	7449 (0.997%)
Day 2	469363	1000	383 (0.0816%)	325 (0.0692%)	69 (0.0147%)	44633 (9.51%)	777 (0.166%)
Day 3	637132	1000	576 (0.090%)	103 (0.0162%)	0	74951 (11.8%)	679 (0.107%)
Day 4	501563	1000	236 (0.047%)	286 (0.057%)	36 (0.00718%)	41812 (8.34%)	588 (0.111%)
Day 5	255121	1000	314 (0.123%)	76 (0.0298%)	0	8800 (3.45%)	390 (0.153%)
Day 6	509049	1000	226 (0.0444%)	87 (0.0171%)	0	8541 (1.68%)	313 (0.0615%)
Day 7	370976	1000	630 (0.170%)	31 (0.00836%)	0	20502 (5.53%)	661 (0.178%)
Day 8	648664	1000	1919 (0.296%)	126 (0.0194%)	0	9087 (1.40%)	2045 (0.315%)
Day 9	574674	1000	697 (0.121%)	618 (0.108%)	0	102382 (17.8%)	1315 (0.229%)
Day 10	563762	1000	163 (0.0289%)	67 (0.0119%)	0	27754 (4.92%)	230 (0.0408%)
Day 12	682304	1000	494 (0.0724%)	206 (0.0302%)	0	5857 (0.858%)	700 (0.103%)

3.2 *Gulf of Mexico Dataset*

The Gulf of Mexico study[10] was an investigation into the distribution of near-surface fish after the Deepwater Horizon oil spill. Of particular interest was any effect on the distribution of flying fish, which were thought to be vulnerable because they pass through the oil floating on the surface. This dataset was selected for this work because flying fish created a large number of individual fish returns.

The survey was conducted over 12 days with images being taken both at night and during the day, using the lidar system described in Sec. 2.1. The data contained instances of flying fish, schools of fish, jellyfish, and plankton layers, which were manually labeled in a previous study.[10] A summary of the Gulf of Mexico dataset is shown in Table 2, and a representative example of lidar data is shown in Fig. 2.

The original data was in the form of 1000 x 1000 pixel PNG images, with multiple images for each day; for each PNG image, object labels were denoted by start and end columns in the image. The original data were transformed such that each day became a separate MATLAB mat file. The labels were placed in a matrix, with each row corresponding to one of the four different objects of interest and each column indicating whether an object was present. To be more consistent with the Yellowstone Lake data, plankton labels were ignored, and the fish, fish school, and jellyfish labels were combined into a single binary label.

4 Methods

Two sets of experiments were run on the Yellowstone Lake and Gulf of Mexico datasets: one with classifiers that were trained on the first day of each dataset and tested on the remaining days, and one with classifiers that were trained on 80% of each dataset and tested on the remaining 20%. The first experiment was intended to simulate a multi-day campaign in which the researchers want to train a classifier after the first day and then use that classifier to predict fish regions during subsequent days. Although the second experiment required more manually labeled data

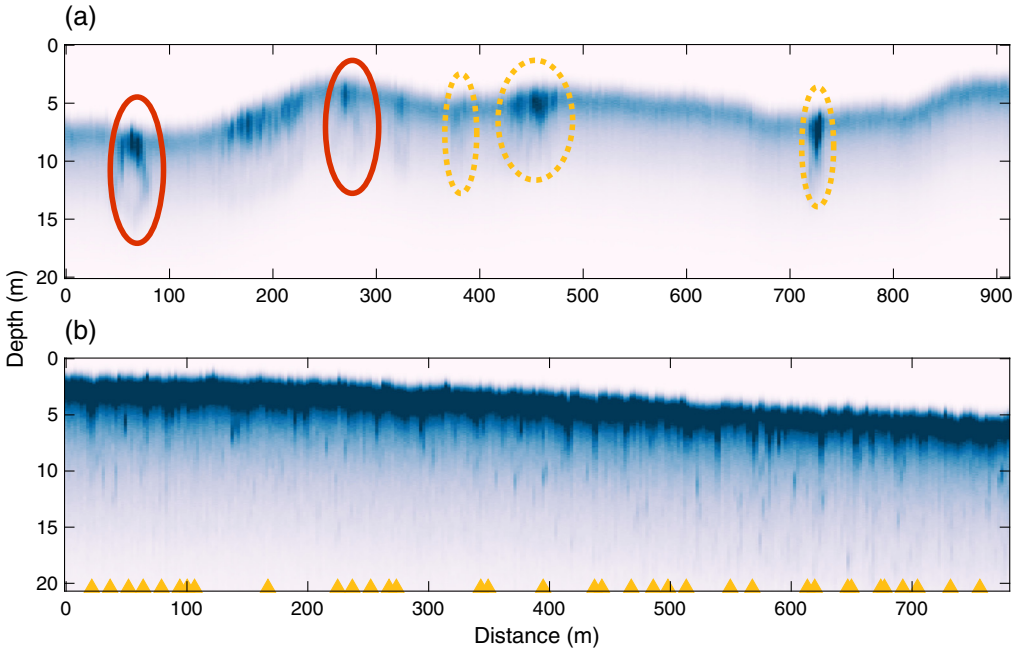

Fig. 2 Example cross-polarized lidar data from the Gulf of Mexico. (a) Data from day 2. Hollow jellyfish aggregations[59] are annotated with red ellipses, and fish schools are highlighted with yellow dotted ellipses. (b) Data from day 1. The yellow triangles indicate lidar shots in which a single fish was manually found and labeled. The colormap's maximum was scaled to increase contrast.

than the first experiment, it can be used to train a classifier for subsequent campaigns in the same location using the same lidar instrument; the second experiment also has the potential to find fish that the manual labels missed.

The following sections describe the algorithms, evaluation metrics, and procedure used in the experiments.

4.1 *Classification Algorithms*

As stated in Sec. 1.2, the following classifiers were used in the experiments: SVM,[38] decision trees,[39] LDA,[40] neural networks,[41] and RUSBoost.[42] SVM, decision trees, and LDA are all common classifiers, so we refer the reader to the corresponding references for more details. The neural network architecture that was used is shown in Fig. 3.

Unlike the other algorithms used in this paper, RUSBoost is an ensemble technique specifically designed for imbalanced learning. RUSBoost introduces random undersampling into the AdaBoost.M2[60] boosting algorithm, with the goal of providing better performance than

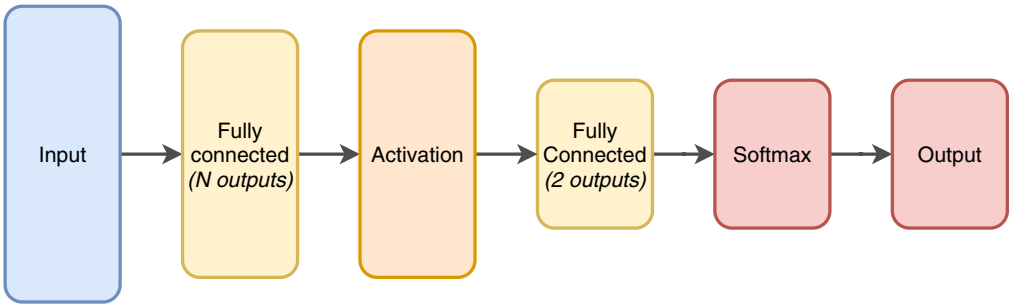

Fig. 3 MATLAB's *fitcnet* neural network architecture. The size of the first fully connected layer and the activation function were tuned, as described in Sec. 4.4.1. The second fully connected layer has 2 outputs, corresponding to the 2 classes—"no fish" and "fish."

undersampling and boosting provide individually. Concretely, RUSBoost randomly undersamples the majority class at the beginning of each iteration of AdaBoost. Random undersampling and boosting are described in the following paragraphs.

4.1.1 *Random undersampling*

Random undersampling[44] is a common method for working with imbalanced datasets, as it is a simple way to help balance class proportions. Given a dataset $S_0 = S_{\text{maj}} \cup S_{\text{min}}$ composed of a majority class $S_{\text{maj}} \subset S_0$ and a minority class $S_{\text{min}} \subset S_0$, random undersampling removes a random subset $S_{\text{remove}} \subset S_{\text{maj}}$ of the majority class to create a new dataset $S_1 \subset S_0 = (S_{\text{maj}} \setminus S_{\text{remove}}) \cup S_{\text{min}}$. Random undersampling's main drawback is that removing examples from the majority class results in loss of information;[61] however, random undersampling has been show to work well in practice.[62] In addition to reducing class imbalance, random undersampling also reduces training time by reducing the dataset size; given the size of the Gulf of Mexico dataset in Sec. 3.2, reducing training time is advantageous.

4.1.2 *Boosting*

Boosting algorithms are iterative algorithms that increase the weights of misclassified points and decrease the weights of correctly classified points after each iteration.[47] This increases the probability that the misclassified examples will be correctly classified in subsequent iterations. AdaBoost[60] is one of the most common boosting techniques. In each iteration, Adaboost trains a weak learner and updates the weights of samples based upon the weak learner's hypothesis. After all iterations are done, the weak learners' hypotheses are weighted and combined to form the final hypothesis.

All of the classifiers used in this study are available in MATLAB R2021a. The classifiers were implemented with the following functions: *fitclinear* for the SVM, *fitcdiscr* for LDA, *fitctree* for the decision tree, *fitcnet* for the neural network, and *fitcensemble* for RUSBoost. The base learner for RUSBoost was a decision tree. By default, MATLAB's implementation of RUSBoost undersamples the majority class to give a perfectly balanced class ratio.

4.2 *Evaluation Metrics*

In this work, classification performance was measured with recall, precision, and the F_3 score, which is a particular combination of recall and precision. All of these metrics range between 0 and 1, with 1 being the best. These metrics can be understood in terms of the confusion matrix, shown in Fig. 4.

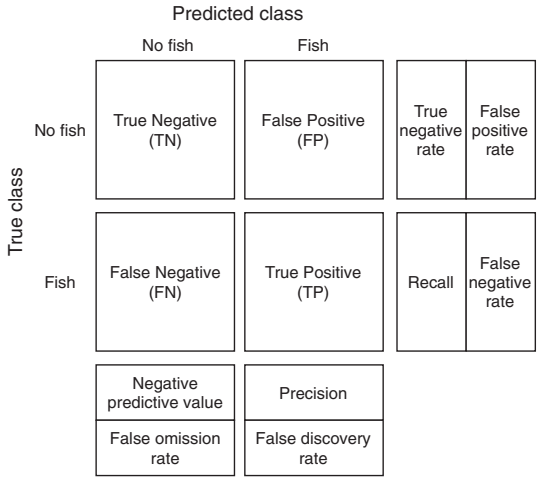

Fig. 4 Confusion matrix terminology. "No fish" is considered the negative class, and "fish" is considered the positive class.

Recall is a measure of how many true fish were correctly predicted, and it is defined as

$$\text{recall} = \frac{TP}{TP + FN}. \tag{1}$$

A recall of 1 means that all fish were identified.

Precision, on the other hand, is a measure of how many identified fish were actually fish. Precision is defined as

$$\text{precision} = \frac{TP}{TP + FP}. \tag{2}$$

A high precision value means the classifier did not mislabel many lidar shots as containing fish, whereas a low precision value means that the classifier mislabeled many non-fish as fish.

Since there are very few fish in the datasets, finding the majority of the fish is more important than accidentally classifying non-fish as fish; however, having a very low precision value is not desirable, as that would mean researchers would have to manually inspect and discard large amounts of unimportant data.

The F_β score is the weighted harmonic mean of precision and recall that weights recall β times higher than precision. Since recall is more important than precision in our application, the F_3 score is used, and it weights recall three times higher than precision. The F_β and F_3 scores are defined as

$$F_\beta = (1 + \beta^2)\frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}, \tag{3}$$

$$F_3 = 10 \cdot \frac{\text{precision} \cdot \text{recall}}{9 \cdot \text{precision} + \text{recall}}. \tag{4}$$

The F_3 score provides a single metric that can be used to evaluate classification performance.

Recall, precision, and F_3 score were computed per-shot and per-region. Per-shot metrics were computed directly using the ground truth and predicted labels for each shot. For the per-region metrics, labels were grouped into 1000-shot windows. For the ground truth labels, a region was considered a region of interest (ROI) if at least one fish label was present. For the predicted labels, the number of fish labels needed for an ROI was tuned as described in Sec. 4.4.1.

4.3 *Preprocessing*

Since this paper focuses on introducing machine learning to fish detection in airborne lidar data, we decided to perform minimal preprocessing on the raw data; most of the preprocessing steps used here are standard practice in previous studies.[8,12] No feature extraction was performed.

Figure 5 shows an overview of the preprocessing procedure. The first two preprocessing steps—surface detection and correction—were performed to compensate for changes in airplane elevation and ensure that a given row in the lidar image nominally corresponded to a given water depth. The depth adjustment step was performed to reduce the dimensionality of the data and to focus on regions most likely to contain meaningful lidar data. Save for the surface detection and smoothing, all processing was performed on the cross-polarized data because it provides better contrast between fish and the background water column.

4.3.1 *Surface detection and smoothing*

The surface of the water in each lidar shot was detected by finding the maximum return in the copolarized channel and then moving up toward the lidar until 25% of the maximum return was reached:

$$s_{\text{raw}}[m] = \max_n\{n | y_n[m] \leq 0.25 \cdot \max(\mathbf{y}[m])\}, \tag{5}$$

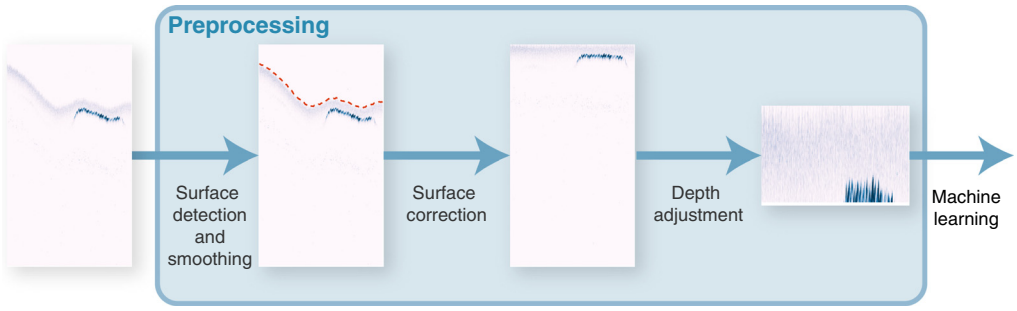

Fig. 5 Data preprocessing overview. The original lidar data (far left) were first run through an algorithm that detected and smoothed the surface of the water (dashed red line). Once the surface was detected, surface correction was applied to compensate for changes in airplane elevation, and data above the surface of the water were removed. Finally, the depth of the image was adjusted based on the lidar's penetration depth or prior knowledge about the expected depth of fish.

where $s_{\text{raw}}[m]$ is the surface index for shot m, $\mathbf{y}[m]$ is the vector of return values for shot m, and $y_n[m]$ is return value at range bin n. This is similar to what was done in a previous study by Roddewig et al.[11] As noted in the previous study,[11] this algorithm might fail when the volumetric scattering from the water below the surface is stronger than the surface return;[63] however, the lidar instruments used in this study are not designed for high-accuracy bathymetry, so an occasional error in surface detection is acceptable. Previous studies[52,64,65] have shown that the surface return preserves polarization, which is why the copolarized channel was used for surface detection.

Upon visual inspection, the surface location detected by Eq. (5) occasionally varied more drastically between adjacent shots than would be expected. To compensate for this, the surface was smoothed with a 10-tap moving average filter:

$$s_{\text{smooth}}[m] = \frac{1}{10}\sum_{l=0}^{9} s_{\text{raw}}[m-l]. \tag{6}$$

A filter length of 10 was chosen visually to give a line that smoothly followed the surface.

The Yellowstone data contained atmospheric returns that were incorrectly detected as the water surface. To overcome this, the first 600 and 512 range bins were skipped during surface detection in the 2015 and 2016 data, respectively. For the sake of simplicity, the number of range bins to skip was chosen via visual inspection for each dataset.

4.3.2 *Surface correction*

Once the surface was found, each lidar shot was shifted up so the surface started at the first row, and then the lidar shot was padded with zeros at the bottom, maintaining the original number of rows. Surface correction was performed to ensure that each row in the image corresponded to a specific distance beneath the surface, regardless of the altitude of the aircraft. Thus the surface correction allowed each row to be interpreted by the machine learning algorithm as a specific feature, in this case, the intensity of the lidar return corresponding to a specific depth in the water.

4.3.3 *Depth adjustment*

After surface correction, the height of the image was reduced based on the penetration depth of the lidar and the expected fish depth. In the Yellowstone Lake data, the image was reduced to 60 rows, which corresponded to a depth of ~8.65 m. The Gulf of Mexico data were reduced to 150 rows, which corresponded to a depth of ~17.3 m. The depth adjustment accomplished three main purposes. First, the adjustment reduced the dimensionality of the problem, which decreases the complexity of the machine learning algorithms. Second, the adjustment considered the physical limitations of the depth penetration of the lidar systems used in this study,[8,21] as well as the

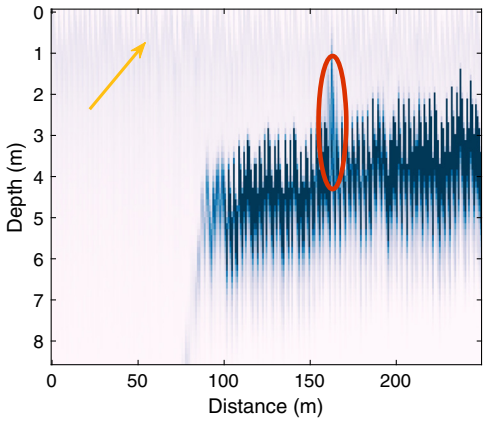

Fig. 6 Lidar data from Fig. 1 after preprocessing. The height-corrected surface of the water is indicated by the yellow arrow, and a fish hit is highlighted by the red ellipse. Radiance values were compressed to increase contrast.

expected physical locations of fish. Finally, truncating the bottom rows ensured that the algorithm was not influenced by the manually padded zeros introduced in the surface correction process. An example of Yellowstone data after preprocessing is shown in Fig. 6.

4.4 *Training Procedure*

After preprocessing was performed, the cross-polarized data were split into training and testing sets. Rather than randomly selecting individual shots for the training and testing sets, the data were split by ROI; we did this for two reasons: (1) since fish often span multiple shots, we did not want to split a single fish hit between the training and testing sets; and (2) splitting the data by ROI allows us to tune the number of predicted labels required for a region to be considered an ROI. The data splitting procedure was the same for both the Yellowstone and Gulf of Mexico datasets.

When creating the training and testing data, each day was split into regions of 1000 adjacent shots. This matches the typical window size used during visual analysis. In general, the last region in each day contained less than 1000 shots because the number of shots in the day was not a multiple of 1000.

For experiments that used the first day as training data, the first day was split by region into 3 folds for cross validation. The folds were randomly selected, and they were stratified such that the class proportions in each fold were approximately the same.

For the experiments that randomly selected the training and testing data, 80% of the regions were randomly selected as training data, while the remaining 20% were put into the testing set. The training and testing sets were stratified such that the class proportions in each were approximately the same. The training set was then split by region into three folds for cross validation. The regions in each fold were randomly selected, and the folds were stratified to maintain class proportions.

4.4.1 *Parameter tuning*

Parameter tuning was performed using three-fold cross validation on the training sets. In addition to tuning each classifier's hyperparameters, we also tuned the undersampling percentage and the number of predicted labels needed for a region to be considered an ROI.

Undersampling. Using the default parameters for each classifier, we tuned the undersampling ratio by performing a grid search between [0, 0.95] in increments of 0.05. For each grid point, the classifiers were trained using three-fold cross validation. The F_3 score was recorded for each fold; the scores were then averaged to give an F_3 score for the grid point. The undersampling ratio that resulted in the maximum F_3 score was chosen for each classifier.

Table 3 Hyperparameter search ranges for the classifiers used in this study. Most of the model-specific parameter search ranges were set to the default search ranges in MATLAB R2021a. $|S|$ indicates the number instances in the dataset. A † indicates that the search values were logarithmically spaced. $\lfloor \cdot \rfloor$ is the floor function.

SVM		LDA		Neural net	
Lambda	$\frac{1}{\lvert S\rvert}[10^{-5}, 10^{5}]^{\dagger}$	Delta	$[10^{-6}, 10^{3}]^{\dagger}$	Layer size	$[10,50] \in \mathbb{Z}$
Regularization	{ridge, lasso}	Gamma	[0, 1]	Activation	{relu, sigmoid, tanh}
FN cost	$[1,20] \in \mathbb{Z}$	FN cost	$[1,20] \in \mathbb{Z}$	—	—
Decision tree		RUSBoost		—	—
Max number of splits	$[1, \lvert S\rvert - 1]^{\dagger} \in \mathbb{Z}$	Learning cycles	$[10,500]^{\dagger} \in \mathbb{Z}$	—	—
Min leaf size	$[1, \lfloor \lvert S\rvert/2 \rfloor]^{\dagger} \in \mathbb{Z}$	Learning rate	$[10^{-3}, 1]^{\dagger}$	—	—
Split criterion	{gdi, deviance}	Max number of splits	$[1, \lvert S\rvert - 1]^{\dagger} \in \mathbb{Z}$	—	—
FN cost	$[1,20] \in \mathbb{Z}$	Min leaf size	$[1, \lfloor \lvert S\rvert/2 \rfloor]^{\dagger} \in \mathbb{Z}$	—	—
—	—	Split criterion	{gdi, deviance}	—	—
—	—	FN cost	$[1,20] \in \mathbb{Z}$	—	—

Model hyperparameters. After tuning the undersampling ratios, we tuned each classifier's hyperparameters using MATLAB's Bayesian optimization function, bayesopt. For classifiers that supported a cost matrix, the cost of missing a fish was also tuned. Table 3 shows the hyperparameters that were tuned for each classifier. We used the following bayesopt settings:

1. IsObjectiveDeterministic = true
2. AcquisitionFunctionName = expected-improvement-plus
3. MaxObjectiveEvaluations = 20

When a classifier achieved a lower F_3 score after hyperparameter tuning, the model parameters were set back to their defaults and only the cost of missing a fish was tuned. When tuning hyperparameters, we undersampled the cross validation training sets by the optimal ratios found via the previous grid search.

ROI label tuning. As discussed in Sec. 5, the classifiers predicted many false positives. Often, the false-positive predictions were in regions that contained fish. When assessing the classifier's ability to correctly predict fish-containing regions of interest, one needs to know when to flag a region as a predicted ROI. Toward this end, we tuned the number of predicted fish labels per region that are needed to flag a region as a true positive. After training the tuned classifiers from the previous steps on the cross validation training sets, the number of positive labels needed per region was swept between 1 and 100. The number of labels that resulted in the highest average F_3 score was chosen for each classifier.

5 Results

Experimental results obtained on both experiments are reported below for both datasets. Due to space considerations, performance metrics are only reported for the ROI results. The per-shot results generally identified extra fish per region; a representative example is shown in Sec. 6.

5.1 *First-Day Training Set*

As was discussed in Sec. 4, the first set of experiments were designed to simulate the real-world scenario of training a classifier after the first flight and then using that classifier during subsequent flights during the same campaign. This section shows the results of those experiments on the Yellowstone Lake and Gulf of Mexico datasets.

5.1.1 *Yellowstone Lake*

Table 4 shows the hyperparameters obtained by following the procedure in Sec. 4.4.1 using the 2015 Yellowstone Lake flight as the training data. The SVM never predicted any fish during hyperparameter tuning, so the default parameter values were used for the final training.

Figure 7 shows the ROI cross validation results obtained on the 2015 flight. The parameters in Table 4 were used when training the classifiers. As seen in Fig. 7, the SVM was uninformative. LDA, the neural network, and RUSBoost properly identified all fish-containing regions. The decision tree only missed one fish-containing region.

The neural network achieved the highest F_3 score during cross validation; consequently, in a real-world scenario, the neural network would be chosen to make predictions during the following days of the campaign. However, as seen in Fig. 8 and Table 5, the neural network did not achieve the highest F_3 score on the testing data (2016 flight). LDA achieved the best F_3 score on the testing data, but achieved the second best score on the training data. As seen in Table 5, LDA did not discard as many regions as the neural network did; compared with LDA, the neural network would have saved manual analysis time in a real-world campaign.

5.1.2 *Gulf of Mexico*

Table 6 shows the hyperparameters obtained using day 1 of the Gulf of Mexico dataset as the training data. Once again, as seen in Fig. 9, the SVM was nearly uninformative. The cross validation performances of the other classifiers, however, are more varied than they were for

Table 4 Hyperparameter values for the first day experiment on Yellowstone Lake.

SVM		LDA		Neural net	
Lambda	9.193×10^{-6}[a]	Delta	0.0007181	Layer size	33
Regularization	ridge[a]	Gamma	0.99884	Activation	relu
FN cost	1[a]	FN cost	20	Undersampling	0.9
Undersampling	0	Undersampling	0.45	# labels for ROI	9
# labels for ROI	1	# labels for ROI	25	—	—
Decision tree		**RUSBoost**			
Max number of splits	2	Learning cycles	11	—	—
Min leaf size	53	Learning rate	0.59087	—	—
Split criterion	gdi	Max number of splits	2	—	—
FN cost	19	Min leaf size	3	—	—
Undersampling	0.95	Split criterion	gdi	—	—
# labels for ROI	43	FN cost	16	—	—
—	—	Undersampling	0.45	—	—
—	—	# labels for ROI	43	—	—

[a]Indicates that the parameter values were left at their default values because hyperparameter tuning did not improve classification results.

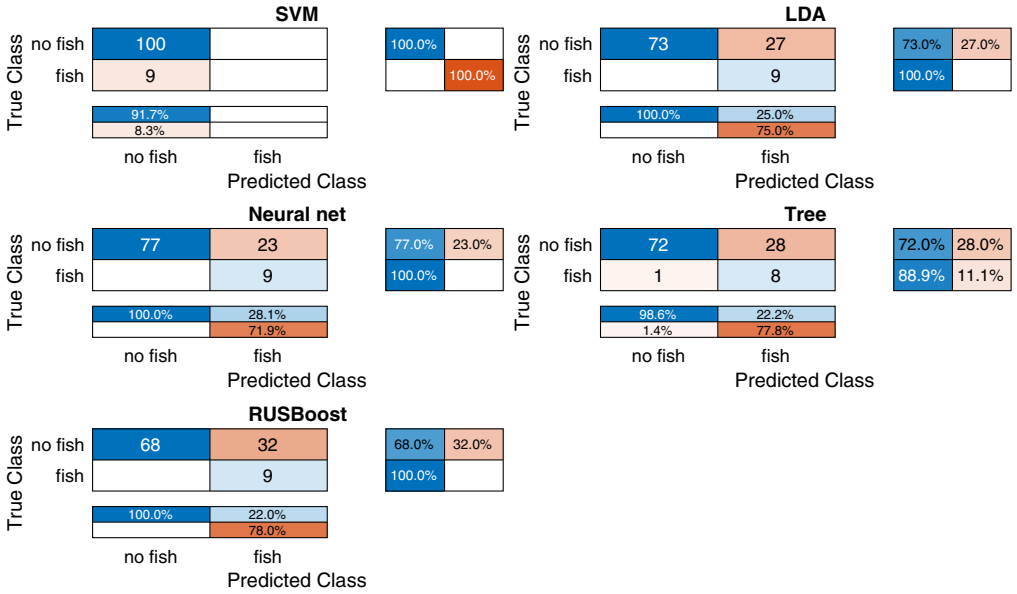

Fig. 7 ROI cross validation results for the 2015 Yellowstone Lake flight.

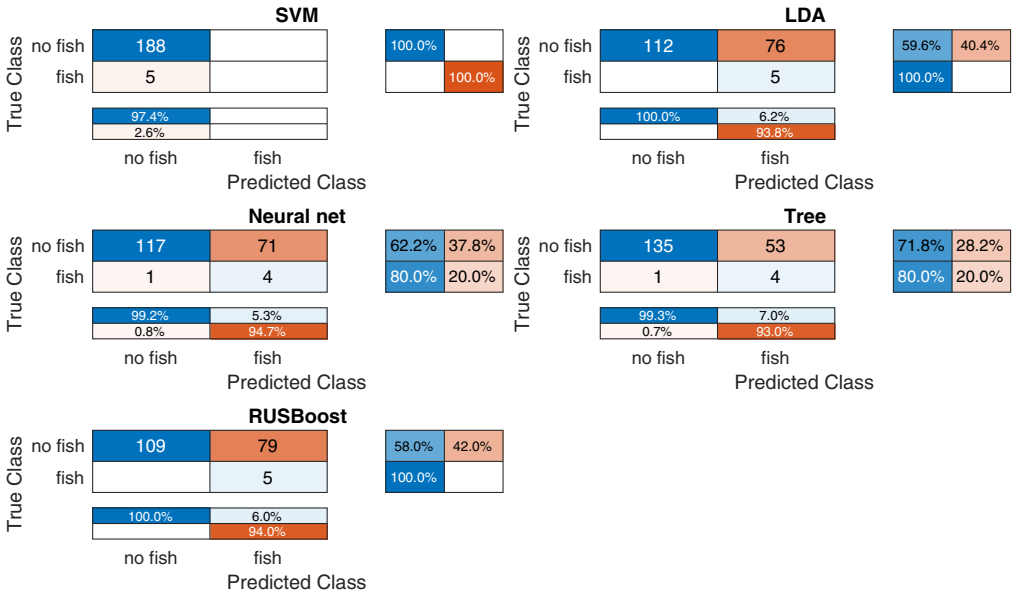

Fig. 8 ROI holdout results for the 2016 Yellowstone Lake flight.

Table 5 F_3 scores achieved on the cross validation and holdout sets for the first day experiment on the Yellowstone Lake dataset. Additionally, the percent of regions that were discarded by each classifier are reported. The best F_3 scores are marked in bold.

	SVM	LDA	Neural network	Decision tree	RUSBoost
Cross validation F_3	0	0.7692	**0.7965**	0.6838	0.7377
Holdout F_3	0	**0.3968**	0.3333	0.3922	0.3876
% regions discarded	100	58.03	61.14	70.47	56.48

Table 6 Hyperparameter values for the first day experiment on Gulf of Mexico dataset.

SVM		LDA		Neural net	
Lambda	8.5538×10^{-8}	Delta	1.1273×10^{-6}	Layer size	42
Regularization	lasso	Gamma	0.81328	Activation	tanh
FN cost	11	FN cost	14	Undersampling	0.95
Undersampling	0	Undersampling	0.8	# labels for ROI	1
# labels for ROI	46	# labels for ROI	1	—	—
Decision tree		**RUSBoost**			
Max number of splits	693	Learning cycles	349	—	—
Min leaf size	347	Learning rate	0.77313	—	—
Split criterion	gdi	Max number of splits	735	—	—
FN cost	20	Min leaf size	4	—	—
Undersampling	0.8	Split criterion	gdi	—	—
# labels for ROI	1	FN cost	16	—	—
		Undersampling	0.8	—	—
		# labels for ROI	1	—	—

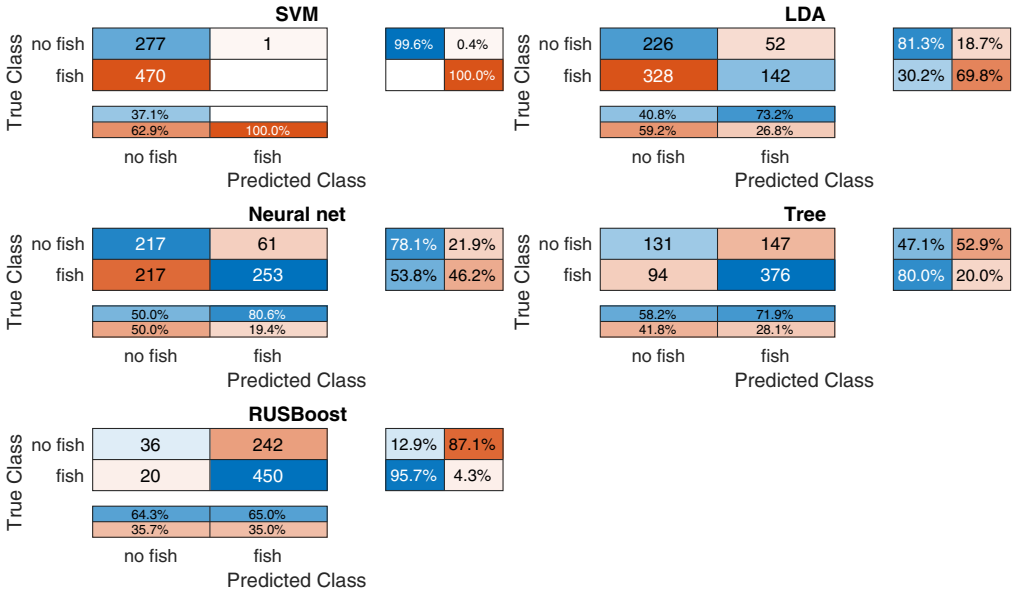

Fig. 9 ROI cross validation results for day 1 of the Gulf of Mexico dataset.

the Yellowstone experiment in Sec 5.1.1. As seen in Fig. 9 and Table 7, RUSBoost achieved the highest F_3 score on the training data; however, this came at the cost of misclassifying 242 (87.1%) regions that did not contain fish. The decision tree, which had the second highest F_3 score, only misclassified 147 (52.9%) regions that did not contain fish. It is worth noting that day 1 contains more fish-containing regions than fishless regions, in spite of the fact that fish make up only 0.997% of the dataset (Table 2).

On the testing set (days 2–12), the decision tree achieved a higher F_3 score than RUSBoost did, as shown in Table 7. This is because the tree achieved a higher precision, which can be seen

Table 7 F_3 scores achieved on the cross validation and holdout sets for the first day experiment on the Gulf of Mexico dataset. Additionally, the percent of regions that were discarded in the testing data by each classifier are reported. The best scores F_3 scores are marked in bold.

	SVM	LDA	Neural network	Decision tree	RUSBoost
Cross validation F_3	0	0.3210	0.5568	0.7911	**0.9143**
Holdout F_3	0	0	0.4354	**0.7983**	0.7419
% regions discarded	100	100	73.1915	12.7660	26.8085

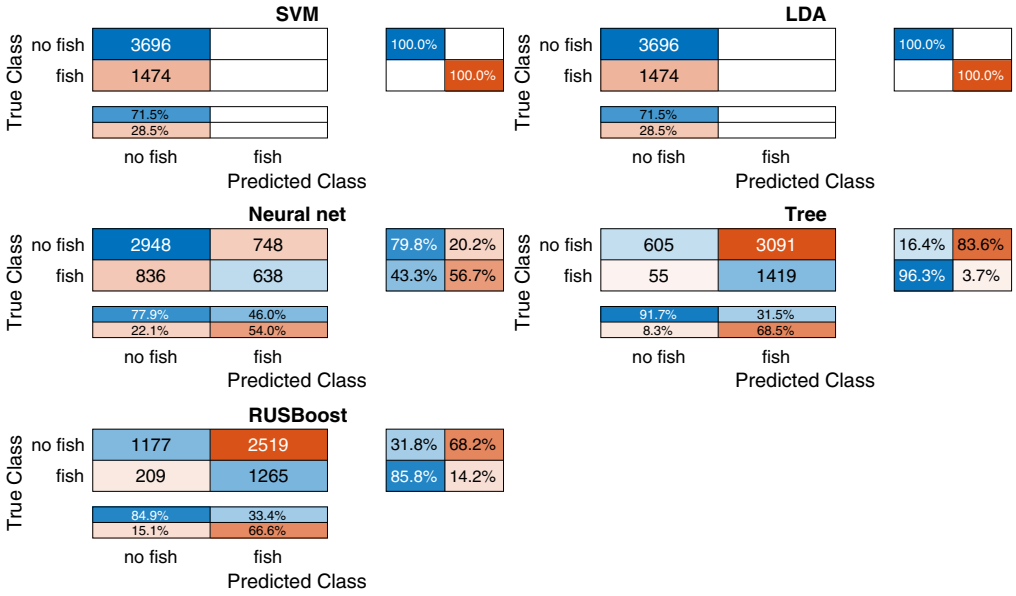

Fig. 10 ROI holdout results for days 2–12 of the Gulf of Mexico dataset.

in Fig. 10; however, this came with the tradeoff of incorrectly identifying an additional 572 regions as containing fish, which would result in more manual analysis time. Ultimately, RUSBoost's performance was not significantly worse than the decision tree's performance and thus would still have been useful during a real-world campaign.

5.2 *Randomized Training Set*

The second set of experiments followed a more traditional approach: the training and test sets were randomly chosen with an 80/20 split. Compared with the "first day" experiments in Sec. 5.1, the training sets for the following experiments contained more data and thus more examples of fish. Generally, training with more data produces more accurate models. The classifiers trained in these experiments could be used in future campaigns that use the same lidar instrument at the same location as the previous campaign. The results of these experiments are reported for both datasets in the following sections.

5.2.1 *Yellowstone Lake*

Table 8 shows the hyperparameters obtained by following the procedure in Sec. 4.4.1. The SVM did not perform better after hyperparameter tuning, so the default parameters were used. RUSBoost also did not perform better after the initial hyperparameter tuning. The RUSBoost hyperparameters were then set to defaults, and the false-negative cost was tuned; this resulted in a higher F_3 score than using default parameters with a false-negative cost equal to 1.

Table 8 Hyperparameter values for the randomized training set experiment on the Yellowstone Lake dataset.

SVM		LDA		Neural net	
Lambda	4.1497×10^{-6}[a]	Delta	0.0014063	Layer size	41
Regularization	ridge[a]	Gamma	0.43906	Activation	tanh
FN cost	1[a]	FN cost	18	Undersampling	0.9
Undersampling	0	Undersampling	0.45	# labels for ROI	16
# labels for ROI	1	# labels for ROI	31	—	—
Decision tree		**RUSBoost**			
Max number of splits	41	Learning cycles	100[a]	—	—
Min leaf size	2	Learning rate	1[a]	—	—
Split criterion	deviance	Max number of splits	1[a]	—	—
FN cost	18	Min leaf size	1[a]	—	—
Undersampling	0.95	Split criterion	gdi[a]	—	—
# labels for ROI	100	FN cost	9	—	—
—	—	Undersampling	0.45	—	—
—	—	# labels for ROI	68	—	—

[a]Indicates that the parameter values were left at their default values because hyperparameter tuning did not improve classification results.

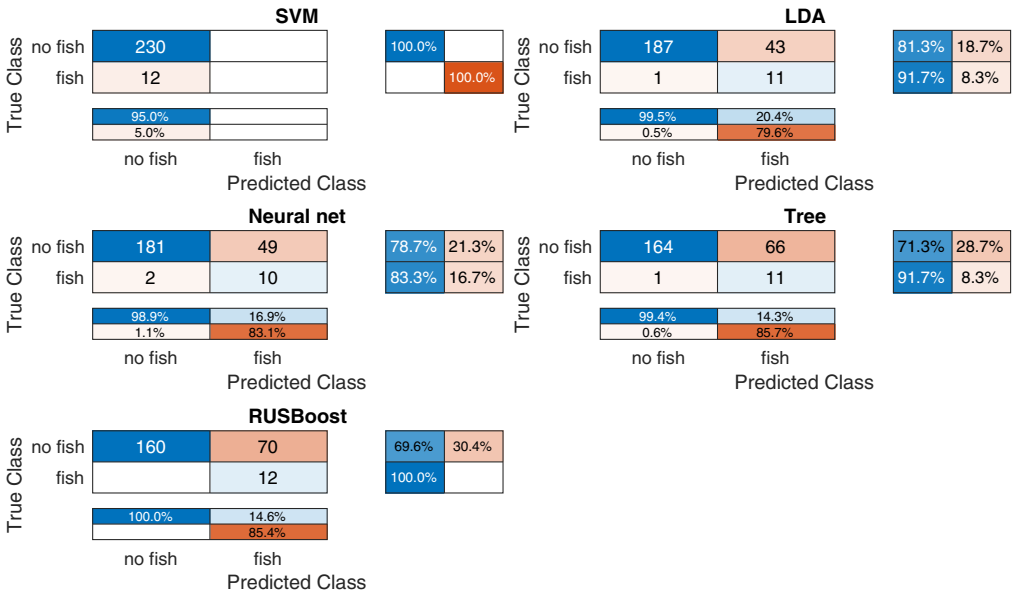

Fig. 11 ROI cross validation results for the Yellowstone Lake randomized training set.

Figure 11 shows the cross validation results obtained using classifiers trained with the parameters in Table 8. As shown in Table 9, LDA achieved the highest F_3 score, followed by RUSBoost. RUSBoost, however, achieved a perfect recall, whereas LDA missed one fish-containing region.

Interestingly, as seen in Fig. 12, LDA missed all fish regions in the testing set, whereas RUSBoost found all of them. The neural network was the only other classifier to find any fish

Table 9 F_3 scores achieved on the cross validation and holdout sets for the randomized train/test split experiment on the Yellowstone Lake dataset. Additionally, the percent of regions that were discarded in the testing data by each classifier are reported. The best F_3 scores are marked in bold.

	SVM	LDA	Neural network	Decision tree	RUSBoost
Cross validation F_3	0	**0.6790**	0.5988	0.5946	0.6316
Holdout F_3	0	0	0.3125	0	**0.5263**
% regions discarded	100	75	76.67	96.67	66.67

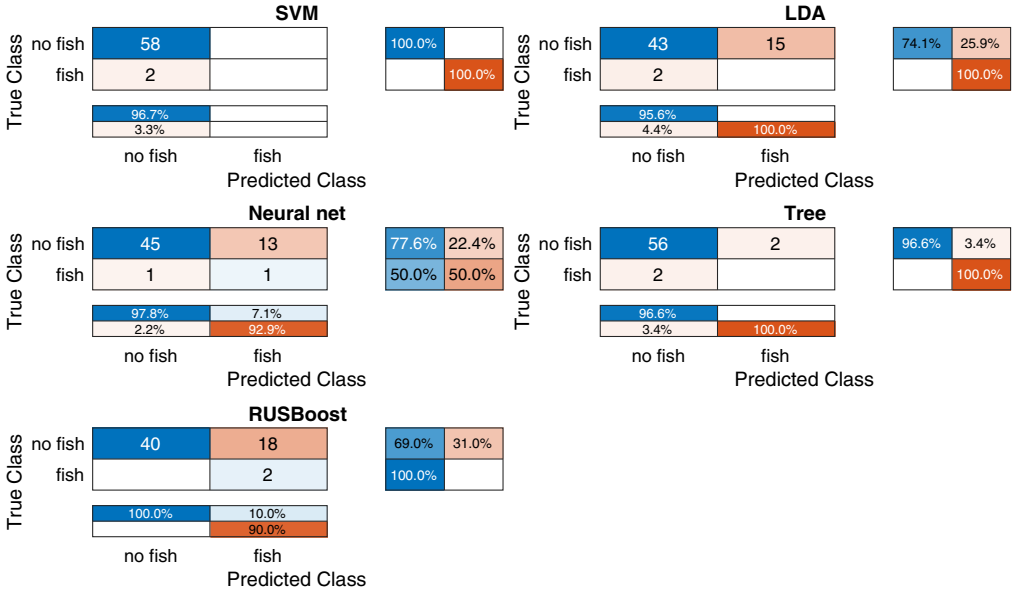

Fig. 12 ROI holdout results for the Yellowstone Lake randomized testing set.

regions. If the best classifier was chosen based purely upon the cross validation F_3 scores, LDA would have been chosen, only to prove unsuccessful during testing.

5.2.2 *Gulf of Mexico*

Table 10 shows the hyperparameters obtained via tuning on the training set. The default LDA hyperparameters performed better than all of the hyperparameters examined during tuning, so only the false-negative cost was tuned. Figure 13 and Table 11 show that the decision tree achieved the highest F_3 score on the training set, while RUSBoost performed second best. As in the first-day experiment on the Gulf of Mexico data, the decision tree's performance came at the cost of marking more fishless regions as containing fish, which would increase the time required for manual inspection.

Looking at the holdout results in Fig. 14 and Table 11, we see that the decision tree outperforms RUSBoost again, with all other classifiers obtaining significantly lower F_3 scores. In line with the cross validation results, the decision tree falsely classified 88.5% of fishless regions in the testing set as containing fish. RUSBoost misclassified 63.3% of fishless regions, while the neural network only misclassified 8.5% of fishless regions. The neural network's low false-positive rate is beneficial for reducing the amount of manual inspection required, but this came with the tradeoff of missing more than half of the fish-containing regions.

Table 10 Hyperparameter values for the randomized training set experiment on the Gulf of Mexico dataset.

SVM		LDA		Neural net	
Lambda	3.8629×10^{-9}	Delta	0[a]	Layer size	44
Regularization	lasso	Gamma	0[a]	Activation	tanh
FN cost	20	FN cost	20	Undersampling	0.95
Undersampling	0	Undersampling	0.8	# labels for ROI	1
# labels for ROI	46	# labels for ROI	1	—	—
Decision tree		RUSBoost		—	—
Max number of splits	4835	Learning cycles	401	—	—
Min leaf size	7	Learning rate	0.79965	—	—
Split criterion	gdi	Max number of splits	200	—	—
FN cost	20	Min leaf size	11	—	—
Undersampling	0.8	Split criterion	Gdi	—	—
# labels for ROI	1	FN cost	6	—	—
—	—	Undersampling	0.8	—	—
—	—	# labels for ROI	1	—	—

[a]Indicates that the parameter values were left at their default values because hyperparameter tuning did not improve classification results.

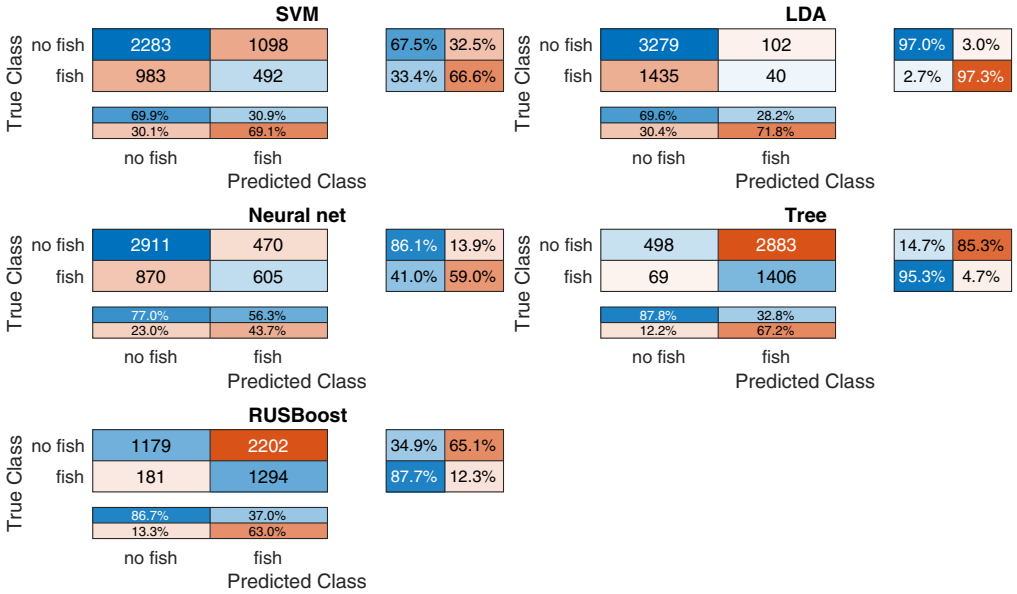

Fig. 13 ROI cross validation results for the Gulf of Mexico randomized training set.

6 Discussion

As seen in Sec. 5, no classifier performed best across all experiments, and the classifier that performed best during cross validation often did not perform best on the testing set. This highlights the fact that classifiers are not guaranteed to perform well on new data; in the context of the lidar data analyzed in this paper, reduced performance on the testing data could be due to the fact

Table 11 F_3 scores achieved on the cross validation and holdout sets for the randomized train/test split experiment on the Gulf of Mexico dataset. Additionally, the percent of regions that were discarded in the testing data by each classifier are reported. The best F_3 scores are marked in bold.

	SVM	LDA	Neural network	Decision tree	RUSBoost
Cross validation F_3	0.3310	0.0298	0.4216	**0.8005**	0.7716
Holdout F_3	0	0.0210	0.3535	**0.8166**	0.7702
% regions discarded	100	98.02	83.84	8.57	29.43

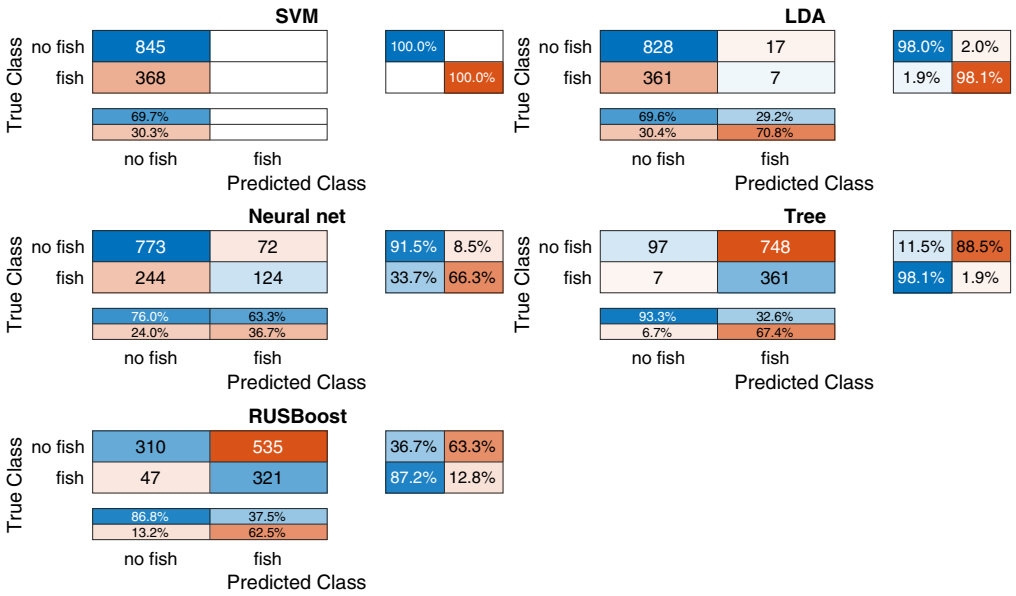

Fig. 14 ROI holdout results for the Gulf of Mexico randomized testing set.

that fish-containing regions often do not have the same characteristics or data distributions as one another. Additionally, the classifiers that we tested traditionally assume that all data instances are independent; due to the nature of our data, this assumption is not entirely true: adjacent instances are spatially correlated. However, in spite of not always achieving similar performance on the testing data, several of the classifiers were able to identify the majority of fish-containing regions in the testing sets.

Although no classifier definitively came out on top, we can see several patterns and make recommendations. On the Yellowstone data, RUSBoost and the neural network performed well across both experiments. While LDA and the decision tree also performed well, the fact that they missed both fish-containing regions in the randomized testing set could be a cause for concern if they were used in future campaigns at Yellowstone Lake.

On the Gulf of Mexico experiments, the decision tree and RUSBoost came out on top in both experiments. The decision tree tended to classify more regions as containing fish than RUSBoost did; on one hand, this increases recall, which is desirable, but it also increases the number of false-positive regions that would need to be inspected. RUSBoost did not find as many fish, but it was able to discard more fishless regions. In a real-world Gulf of Mexico campaign, the choice between using RUSBoost or a decision tree would rely on the relative importance of finding all fish versus reducing manual labor. Furthermore, if one is most interested in reducing manual labor, and can tolerate missing a significant number of fish-containing regions, the neural network would be a good choice because it consistently discarded over 60% of the training data in both experiments on both datasets.

Table 12 Training runtimes in seconds. The runtimes are totals for the entire tuning procedure described in Sec. 4.4.1. The training was run on a Linux computer with an i9-9900k and 32 gigabytes of RAM.

	Yellowstone Lake		Gulf of Mexico	
	First day training time	80% split training time	Fist day training time	80% split training time
SVM	30.38	63.78	129.55	670.96
LDA	16.83	18.73	173.62	935.13
Neural Net	334.74	883.19	9377.30	46401.00
Decision tree	22.64	45.99	578.54	4684.90
RUSBoost	177.80	235.19	2888.80	8074.30

Overall, based on the results seen in this study, we would recommend using either RUSBoost, a decision tree, or a neural network to detect fish regions in aerial lidar surveys. We cannot recommend using an SVM, as it was almost always uninformative; this is likely caused by the high class imbalance in fishery surveys. If reduced training time is important, one should consider using a decision tree; RUSBoost and neural networks had longer training times, as shown in Table 12. It is worth nothing that regardless of which classifier is used, the classifier needs to be trained on data from the specific survey being analyzed; in general, the results from classifiers trained on data from different locations or different instruments will not transfer to new locations or instruments.

Comparing the results between the first-day training set (Sec. 5.1) and the larger, randomized training set (Sec. 5.2), we see that using the larger training set produced better results on the testing set, as expected. The cross validation results on the randomized training set were lower than those on the first-day training set; this makes sense because the randomized training sets were more diverse, resulting in harder-to-learn data distributions. Nonetheless, the results show that our methods can be useful in both real-world scenarios simulated in Secs. 5.1 and 5.2.

Figure 15 shows a representative example of the predicted labels for an ROI. The predicted labels found one fish, missed one fish, and predicted several false fish. This behavior should prompt a human to take a closer look, which is why it was labeled as an ROI.

As an example of the time savings that our method can achieve, consider RUSBoost's holdout results on days 2–12 in Fig. 14. RUSBoost predicted 1386 (26.8%) regions as not containing fish. 1386 regions is ~13.5 h of lidar data. With an average analysis time of 15 min for an hour of lidar data, 1386 regions would take ~4.6 h to analyze. Inspecting the Gulf of Mexico data took about 14 h total; saving 4.6 h of manual labor is an ~33% time savings.

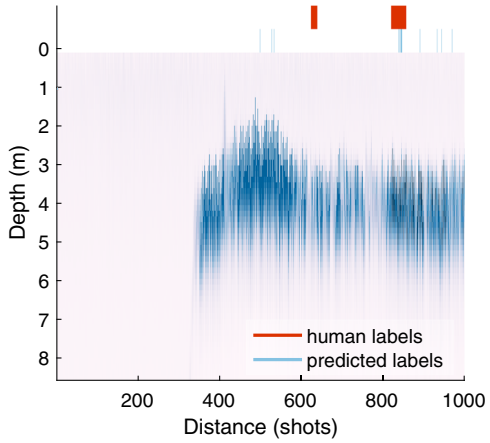

Fig. 15 An example ROI from the 2016 Yellowstone flight.

7 Conclusion

In this study, we demonstrated the feasibility of using supervised machine learning techniques to detect regions that have a high probability of containing fish. We devised two experiments to simulate real-world scenarios: one with a classifier that is trained after the first day of a campaign and used during the subsequent days, and one with a classifier that is trained on a previous campaign in preparation for a new campaign at the same location. In both cases, we found that several classifiers were able to correctly detect the majority of fish-containing regions while reducing the amount of data that would require manual inspection. This is a significant improvement over manual inspection, which is the current state-of-the-art in the field. Additionally, we made recommendations on which classifiers should be used for future lidar-based fishery surveys.

7.1 *Future Work*

Given the plethora of machine learning techniques available, there are many avenues for future work. One of the primary directions will be to extract features from the data, as feature engineering might improve results compared with using the raw data. Using classifiers, such as recurrent neural networks, that model the time-dependency between samples is also a promising direction for further research. Another interesting area of future work is to aggregate multiple classifiers into an ensemble. For example, aggregating the outputs of the RUSBoost, decision tree, and neural network classifiers used in this study could result in improved performance.

In addition to further exploration of supervised learning techniques, we will direct future research efforts toward real-time and unsupervised fish detection methods. Real-time methods would allow researchers to detect fish during flight, enabling personnel on the ground to carry out further operations, e.g., collecting more data or removing invasive species. Unsupervised methods would eliminate the need for ground truth labels, resulting in further time savings. Although there is still room for improvement, our results motivate further application of machine learning techniques in fishery surveys and other lidar-based remote sensing applications.

Acknowledgments

We gratefully acknowledge funding from the U.S. Air Force Research Laboratory through a subcontract with S2 Corp. Additionally, we thank the reviewers and editors for their many insightful comments and suggestions that have greatly improved the quality of this manuscript.

Code, Data, and Materials Availability

The authors are coordinating with data librarians at Montana State University to archive the Yellowstone and Gulf of Mexico fish lidar datasets. The machine learning software as been published on GitHub and archived via Zenodo and assigned a Digital Object Identifier (DOI). The software can be reached at https://doi.org/10.5281/zenodo.5097031. The Zenodo archive will be updated to include a link to the raw data when it becomes publicly available. Until then, the data is available upon request.

References

1. R. A. Myers and B. Worm, "Rapid worldwide depletion of predatory fish communities," *Nature* **423**(6937), 280–283 (2003).
2. J. Hampton et al., "Decline of pacific tuna populations exaggerated?" *Nature* **434**(7037), E1–E2 (2005).
3. T. Polacheck, "Tuna longline catch rates in the Indian Ocean: did industrial fishing result in a 90% rapid decline in the abundance of large predatory species?" *Marine Policy* **30**(5), 470–482 (2006).
4. R. Bauer et al., "Aerial surveys to monitor bluefin tuna abundance and track efficiency of management measures," *Mar. Ecol. Prog. Ser.* **534**, 221–234 (2015).

5. M. Lutcavage, S. Kraus, and W. Hoggard, "Aerial survey of giant bluefin tuna, Thunnus thynnus, in the great Bahama Bank, Straits of Florida, 1995," *Fishery Bull.* **95**, 300–310 (1997).
6. J. H. Churnside, A. F. Sharov, and R. A. Richter, "Aerial surveys of fish in estuaries: a case study in Chesapeake Bay," *ICES J. Mar. Sci.* **68**, 239–244 (2011).
7. J. H. Churnside, J. J. Wilson, and V. V. Tatarskii, "Lidar profiles of fish schools," *Appl. Opt.* **36**, 6011 (1997).
8. J. H. Churnside, J. J. Wilson, and V. V. Tatarskii, "Airborne lidar for fisheries applications," *Opt. Eng.* **40**, 406 (2001).
9. J. Churnside, "A comparison of lidar and echosounder measurements of fish schools in the gulf of mexico," *ICES J. Mar. Sci.* **60**, 147–154 (2003).
10. J. Churnside et al., "Surveying the distribution and abundance of flying fishes and other epipelagics in the northern Gulf of Mexico using airborne lidar," *Bull. Mar. Sci.* **93**, 591–609 (2017).
11. M. R. Roddewig et al., "Dual-polarization airborne lidar for freshwater fisheries management and research," *Opt. Eng.* **56**, 031221 (2017).
12. M. R. Roddewig et al., "Airborne lidar detection and mapping of invasive lake trout in Yellowstone Lake," *Appl. Opt.* **57**, 4111 (2018).
13. W. G. Pichel et al., "Marine debris collects within the north pacific subtropical convergence zone," *Mar. Pollut. Bull.* **54**, 1207–1211 (2007).
14. T. S. Veenstra and J. H. Churnside, "Airborne sensors for detecting large marine debris at sea," *Mar. Pollut. Bull.* **65**, 63–68 (2012).
15. J. H. Churnside and P. L. Donaghay, "Thin scattering layers observed by airborne lidar," *ICES J. Mar. Sci.* **66**, 778–789 (2009).
16. J. H. Churnside and R. D. Marchbanks, "Subsurface plankton layers in the arctic ocean," *Geophys. Res. Lett.* **42**, 4896–4902 (2015).
17. K. Li et al., "A dual-wavelength ocean lidar for vertical profiling of oceanic backscatter and attenuation," *Remote Sens.* **12**, 2844 (2020).
18. W. Li et al., "Influence of characteristics of micro-bubble clouds on backscatter lidar signal," *Opt. Express* **17**(20), 17772–17783 (2009).
19. J. H. Churnside, "Lidar signature from bubbles in the sea," *Opt. Express* **18**, 8294 (2010).
20. M. R. Roddewig, J. H. Churnside, and J. A. Shaw, "Airborne lidar detection of an underwater thermal vent," *J. Appl. Remote Sens.* **11**, 036014 (2017).
21. M. R. Roddewig, J. H. Churnside, and J. A. Shaw, "Lidar measurements of the diffuse attenuation coefficient in Yellowstone lake," *Appl. Opt.* **59**(10), 3097–3101 (2020).
22. S. Rodier et al., "Calipso lidar measurements for ocean sub-surface studies," in *34th Int. Symp. Remote Sens. Environ.* (2011).
23. J. Churnside, B. McCarty, and X. Lu, "Subsurface ocean signals from an orbiting polarization lidar," *Remote Sens.* **5**, 3457–3475 (2013).
24. M. J. Behrenfeld et al., "Space-based lidar measurements of global ocean carbon stocks," *Geophys. Res. Lett.* **40**, 4355–4360 (2013).
25. X. Lu et al., "Ocean subsurface studies with the CALIPSO spaceborne lidar," *J. Geophys. Res. Oceans* **119**, 4305–4317 (2014).
26. M. J. Behrenfeld et al., "Annual boom–bust cycles of polar phytoplankton biomass revealed by space-based lidar," *Nat. Geosci.* **10**, 118–122 (2017).
27. Y. Hu and P. Zhai, "Development and validation of the CALIPSO ocean subsurface data," in *IEEE Int. Geosci. and Remote Sens. Symp.*, IEEE (2016).
28. Y. Hu et al., "Ocean lidar measurements of beam attenuation and a roadmap to accurate phytoplankton biomass estimates," *EPJ Web Conf.* **119**, 22003 (2016).
29. C. A. Hostetler et al., "Spaceborne lidar in the study of marine systems," *Annu. Rev. Mar. Sci.* **10**, 121–147 (2018).
30. D. Dionisi et al., "Seasonal distributions of ocean particulate optical properties from spaceborne lidar measurements in mediterranean and black sea," *Remote Sens. Environ.* **247**, 111889 (2020).
31. H.-G. Maas et al., "Improvements in lidar bathymetry data analysis," *ISPRS—Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **XLII-2/W10**, 113–117 (2019).

32. N. Xu et al., "A method to derive bathymetry for dynamic water bodies using ICESat-2 and GSWD data sets," *IEEE Geosci. Remote Sens. Lett.*, 1–5 (2020).
33. X. Zhao et al., "Improved waveform decomposition with bound constraints for green waveforms of airborne LiDAR bathymetry," *J. Appl. Remote Sens.* **14**, 027502 (2020).
34. J. H. Churnside and J. A. Shaw, "Lidar remote sensing of the aquatic environment: invited," *Appl. Opt.* **59**(10), C92–C99 (2020).
35. J. H. Churnside, E. Tenningen, and J. J. Wilson, "Comparison of data-processing algorithms for the lidar detection of mackerel in the Norwegian sea," *ICES J. Mar. Sci.* **66**(6), 1023–1028 (2009).
36. J. H. Churnside and R. E. Thorne, "Comparison of airborne lidar measurements with 420 kHz echo-sounder measurements of Zooplankton," *Appl. Opt.* **44**(26), 5504 (2005).
37. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., ch. 12.2 and 14.1, Springer Nature, New York (2009).
38. V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer Science & Business Media, New York (2006).
39. L. Breiman et al., *Classification and Regression Trees*, CRC Press, Boca Raton, Florida (1984).
40. R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics* **7**(2), 179–188 (1936).
41. K. Mehrotra, C. K. Mohan, and S. Ranka, *Elements of Artificial Neural Networks*, MIT Press, Cambridge, Massachusetts (1997).
42. C. Seiffert et al., "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Trans. Syst. Man Cybern. Part A* **40**, 185–197 (2010).
43. N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *ACM Sigkdd Explorations Newslett.* **6**(1), 1–6 (2004).
44. H. He and E. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.* **21**(9), 1263–1284 (2009).
45. G. Wu and E. Y. Chang, "Class-boundary alignment for imbalanced dataset learning," in *ICML Workshop Learn. Imbalanced Data Sets II*, Washington, DC, pp. 49–56 (2003).
46. F. Provost, "Machine learning from imbalanced data sets 101," in *Proc. AAAI Workshop Imbalanced Data Sets*, AAAI Press, Vol. 68, pp. 1–3 (2000).
47. G. M. Weiss, "Mining with rarity," *ACM SIGKDD Explorations Newslett.* **6**, 7–19 (2004).
48. S. J. Purkis and J. C. Brock, "LiDAR overview," in *Coral Reef Remote Sensing*, J. Goodman, S. Purkis, and S. Phinn, Eds., pp. 115–143, Springer, Netherlands (2013).
49. F. E. Hoge et al., "Airborne lidar detection of subsurface oceanic scattering layers," *Appl. Opt.* **27**(19), 3969–3977 (1988).
50. Y. A. Goldin et al., "Results of Barents Sea airborne lidar survey," *Proc. SPIE* **6615**, 66150E (2007).
51. V. S. Shamanaev, "Laser sensing of the atmosphere and upper layer of the ocean by an airborn and shipborne lidars," *Russ. Phys. J.* **56**(7), 813–821 (2013).
52. A. P. Vasilkov et al., "Airborne polarized lidar detection of scattering layers in the ocean," *Appl. Opt.* **40**(24), 4353–4364 (2001).
53. H. Liu et al., "Subsurface plankton layers observed from airborne lidar in Sanya Bay, South China Sea," *Opt. Express* **26**(22), 29134–29147 (2018).
54. J. Ma et al., "Compact dual-wavelength blue-green laser for airborne ocean detection lidar," *Appl. Opt.* **59**(10), C87–C91 (2020).
55. H. M. Zorn, J. H. Churnside, and C. W. Oliver, "Laser safety thresholds for cetaceans and pinnipeds," *Mar. Mammal Sci.* **16**(1), 186–200 (2000).
56. L. R. Kaeding, G. D. Boltz, and D. G. Carty, "Lake trout discovered in Yellowstone lake threaten native cutthroat trout," *Fisheries* **21**(3), 16–20 (1996).
57. J. R. Ruzycki, D. A. Beauchamp, and D. L. Yule, "Effects of introduced lake trout on native cutthroat trout in Yellowstone lake," *Ecol. Appl.* **13**(1), 23–37 (2003).
58. T. M. Koel et al., "Nonnative lake trout result in Yellowstone cutthroat trout decline and impacts to bears and anglers," *Fisheries* **30**(11), 10–19 (2005).
59. J. H. Churnside et al., "Hollow aggregations of moon jellyfish (Aurelia spp.)," *J. Plankton Res* **38**, 122–130 (2016).

60. Y. Freund et al., "Experiments with a new boosting algorithm," in *ICML*, Vol. 96, pp. 148–156, Citeseer (1996).
61. G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newslett.* **6**(1), 20–29 (2004).
62. J. V. Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proc. 24th Int. Conf. Mach. Learn.*, ACM Press (2007).
63. G. C. Guenther, "Wind and nadir angle effects on airborne lidar water 'surface' returns," *Proc. SPIE* **0637**, 277–286 (1986).
64. G. M. Krekov, M. M. Krekova, and V. S. Shamanaev, "Laser sensing of a subsurface oceanic layer. II. Polarization characteristics of signals," *Appl. Opt.* **37**(9), 1596–1601 (1998).
65. J. H. Churnside, "Polarization effects on oceanographic lidar," *Opt. Express* **16**(2), 1196–1207 (2008).

Trevor C. Vannoy received his BS and MS in electrical engineering from Montana State University, where he is currently pursuing a PhD in electrical engineering. His research interests include signal processing, machine learning, field-programmable gate arrays, and real-time algorithms for embedded systems.

Jackson Belford received his BS in electrical engineering from Montana State University in spring 2021. He is currently working at Raytheon in Tuscon, Arizona.

Joseph N. Aist received his BS in electrical engineering from Montana State University in spring 2021. He is currently pursuing an MS in electrical engineering at Montana State University.

Kyle R. Rust worked as an undergraduate research assistant for Dr. Bradley Whitaker from January of 2020 to April of 2021. He graduated from Montana State University in April of 2021 with a BS in computer engineering, and he will continue his studies at Northern Arizona University starting in the fall of 2021.

Michael R. Roddewig is a senior research engineer in the Optical Technology Center (OpTeC) at Montana State University. He received his BS from Michigan Technological University, his MS from Colorado School of Mines, and his PhD from Montana State University.

James H. Churnside retired from the NOAA Chemical Sciences Laboratory after 34 years of service in 2019. Since then, he has been working at the Chemical Sciences Laboratory part time as a research scientist with the Cooperative Institute for Research in the Environmental Sciences, which is a joint institute between NOAA and the University of Colorado. For the last 20 years, his primary research interest has been in the application of airborne lidar for fisheries and oceanographic research.

Joseph A. Shaw is the director of the Optical Technology Center and Distinguished professor in the Norm Asbjornson College of Engineering at Montana State University in Bozeman, Montana, USA. He earned his PhD and MS in optical sciences at the University of Arizona, MS in electrical engineering at the University of Utah, and BS in electrical engineering at the University of Alaska Fairbanks. Recognition for his contributions to optics research and education include the Presidential Early Career Award for Scientists and Engineers, the Vaisala Award from the World Meteorological Organization, and the G. G. Stokes Award from SPIE. Shaw is a fellow of both the Optical Society of America (OSA) and SPIE.

Bradley M. Whitaker received his BS in electrical engineering from Brigham Young University and his MS and PhD in electrical and computer engineering from the Georgia Institute of Technology. Whitaker is now an assistant professor at Montana State University in Bozeman, Montana. His research focuses on signal processing and machine learning, with applications in healthcare, military surveillance, and space exploration.