Journal of
Marine Science and Engineering

MDPI

Article

Feature Extraction and Classification of Simulated Monostatic Acoustic Echoes from Spherical Targets of Various Materials Using Convolutional Neural Networks

Bernice Kubicek [1,*], Ananya Sen Gupta [1] and Ivars Kirsteins [2]

[1] Electrical and Computer Engineering, University of Iowa, Iowa City, IA 52242, USA
[2] Naval Undersea Warfare Center, Newport, RI 02841, USA
* Correspondence: bernice-kubicek@uiowa.edu

Abstract: Active sonar target classification remains an ongoing area of research due to the unique challenges associated with the problem (unknown target parameters, dynamic oceanic environment, different scattering mechanisms, etc.). Many feature extraction and classification techniques have been proposed, but there remains a need to relate and explain the classifier results in the physical domain. This work examines convolutional neural networks trained on simulated data with a known ground truth projected onto two time-frequency representations (spectrograms and scalograms). The classifiers were trained to discriminate the target material type, geometry, and internal fluid filling, while the hyperparameters were tuned to the classification task using Bayesian optimization. The trained networks were examined using an explainable artificial intelligence technique, gradient-weighted class activation mapping, to uncover the informative features used in discrimination. This analysis resulted in visual representations that allowed the CNN choices to be related to the physical domain. It was found that the scalogram representation provided a negligible classification accuracy increase compared with the spectrograms. Networks trained to discriminate between target geometries resulted in the highest accuracy, and the networks trained to discriminate the internal fluid of the target resulted in the lowest accuracy.

Keywords: automatic target recognition; continuous wavelet transform; convolutional neural network; elastic wave classification; explainable artificial intelligence

Citation: Kubicek, B.; Sen Gupta, A.; Kirsteins, I. Feature Extraction and Classification of Simulated Monostatic Acoustic Echoes from Spherical Targets of Various Materials Using Convolutional Neural Networks. *J. Mar. Sci. Eng.* **2023**, *11*, 571. https://doi.org/10.3390/jmse11030571

Academic Editors: Tracianne B Neilsen and Haiqiang Niu

Received: 1 February 2023
Revised: 23 February 2023
Accepted: 2 March 2023
Published: 7 March 2023

1. Introduction

There are two types of sonar systems: passive and active. The former is when a hydrophone is recording sound within the ocean, and the latter occurs when a pulse of sound, or a ping, has been sent out to a target of interest in an attempt to determine a target's information. This research is specific to active sonar. Active sonar target recognition and classification has numerous maritime applications, such as harbor monitoring, autonomous underwater vehicle vision, and seabed characterization. However, classification suffers from feature uncertainties due to unpredictable or unknown environmental (salinity, temperature, sound speed profile, etc.) and target parameters (size, shape, orientation, etc.) [1]. Different forms of obstructions, such as fish or bubbles, or oceanic noise may also be present within a sonar's return path and can further entangle a received response [2]. These effects combine and degrade the target-specific informative features used for discrimination.

Machine learning algorithms are commonly used to perform classification of sonar data [3–11]. Many of these classification pipelines employ convolutional neural networks (CNNs). Williams demonstrated classification of sonar images using a 10 layer CNN [3]. Wang et al. used weights found with a deep belief network and then replaced the randomly initialized CNN weights to perform classification of various sonar images [12]. CNNs were used for feature extraction rather than classification by Zhu et al., who used AlexNet to extract sonar image features prior to classification using a support vector machine [7].

Sonar image feature extraction through edge detection was additionally proposed by Wang et al., who created three different CNNs with skip connections and demonstrated its ability to find continuous edges [13]. However, many of these networks are deep and rely on a large number of samples for training. A large amount of public domain experimental field data for training is challenging and costly to obtain [9]. Many approaches, such as feature engineering or extraction [10], transfer learning [4], pretrained networks [6,7], synthetic data generation [4–6], and employing many tiny classifiers [11], have been used to mitigate this challenge. This work is a combination of synthetic data generation, two time-frequency representations, and the employment of moderately sized classifiers that have been optimized for various sonar target recognition tasks. The signals used throughout this work are simulated from known models, giving complete control over the ground truth and simulation options. This allows interpretation of the informative features used by the classifier for discrimination to be related back to the physical domain.

Signal segmentation is performed prior to the projection of simulated target backscattered responses onto two two-dimensional representations. The use of two-dimensional (2D) analysis in conjunction with machine learning techniques is common within the underwater community [10,14–19]. An in-depth review of the feature extraction and classification methods is provided in [20]. Choi et al. employed cross-spectral density matrices as inputs to a variety of classifiers trained to discriminate against submerged or surface ships. In most cases, CNNs provided increased classification, with a lowest binary misclassification rate of 0.92% [15]. Power cepstrums were used as CNN inputs trained to perform detection and ranging of vessels in a variety of SNRs [14]. Other reported research describes a novel chirp wavelet [16] using a three-channel spectrogram CNN classfier [17], or using binary features extracted from acoustic spectra [10,18]. An interesting approach was taken by Luo et al. They used multiple spectrograms at various resolutions as a three-channel input in conjunction with a ResNet-inspired network, achieving up to a 96.32% classification accuracy in ship noise classification [19]. This research differs in the use of simulated signals with a known ground truth, a comparison of the short-time Fourier transform and the continuous wavelet transform representations and their impact on classification, and examination of the classifiers post training to explain the classifier choices.

Deep learning techniques are considered to be state of the art and provide increased classification when compared with traditional approaches [21]. However, these networks suffer from a lack of explainability and interpretability of their results due to their complexity [22]. This creates a lack of trust and transparency in a classifier and is sometimes referred to as a 'black box'. When incorrect classification may result in harmful real-world outcomes (as is the case in sonar classification), there exists the need for explainable artificial intelligence (XAI) [23]. Explainable artificial intelligence (XAI) is an emerging area of study aimed at increasing the interpretability of machine learning choices. XAI attempts to build transparency and increase trust in a classifier algorithm by making the classifier choices interpretable to humans [24]. There are many different approaches to XAI, and the one employed in this work is gradient-weighted class activation mapping (Grad-CAM), which uses the gradients of test images sent through the trained network to determine which feature is the most important one [25]. This technique can be used to find discriminative, class-specific features and has been used to explore trained networks in the medical field [26,27].

Spectrograms and scalograms are two-dimensional representations that describe how the frequency content of a signal changes over time. These representations have been chosen as they are two standard representations that are commonly used. The goal of this research is not to find the optimum time-frequency distributions for classification but rather to interpret the classifier choices and relate the decisions back to the physical domain. A spectrogram has a fixed window size that forces a constant time-frequency resolution, while scalograms have a scaling parameter that allows the contraction and dilation of the window. This provides a varied resolution. The fixed resolution results in spectrograms being an adequate representation of stationary signals and a poor representation of non-stationary

signals. The variable resolution associated with scalograms allows them to be a good representation of non-stationary signals. A comparison between the two representations through classification accuracy is reported within this work. This work is a continuation and expansion of the comparison of spectrogram and scalogram representations of the simulated backscattered responses reported in [28]. This research differs from the previous iteration by updating the signal segmentation, the inclusion of additional simulated targets, the usage of a convolutional neural network that has been optimized for classification, and the examination of the trained classifier to explain its choices and relate them to the physical domain.

A large amount of experimental sonar field data for training is challenging and costly to obtain, but advances in feature extraction and machine learning have been creatively mitigating this problem. However, classifiers lack interpretation to the physical domain. This work uses simulated data that have a known ground truth and a CNN for classification. The hyperparameters are tuned using Bayesian optimization. After training, the networks are examined to determine the important features used by the network for classification. This work is an examination and comparison of two common time-frequency representations, both preceding and following classification. The networks were trained to perform classification for a variety of tasks in order to determine any dependencies between the classification task and optimized hyperparameters. The post-classification examination is performed using an XAI technique to reveal the target-specific features that a CNN uses for discrimination. The key contribution of this work is in the examination of the networks trained on common time-frequency representations and the explanation of the network choices in the physical domain. This post hoc analysis allows for informed decisions to be made for future classification pipelines to intuitively bias networks by forcing them to prioritize influential features. The analysis can additionally be used to examine the failure modes of classifiers and create mitigation strategies.

2. Materials and Methods

In this section, the simulated data generated for this research are described first, followed by the two-dimensional feature representations and classifier. The classification metrics used to compare the performance are reported, and finally an explainable artificial intelligence technique is described. This research was coded and developed using MATLAB 2022a v9.12.0 with the Deep Learning Toolbox [29].

2.1. Data Generation

Elastic targets have both rigid and elastic scattering. The rigid scattering is typically associated with specular or geometric reflections, while the latter is due to the elastic material properties of the target. The common reflections associated with solid elastic spheres are the specular or geometric reflection and the Rayleigh and whispering gallery surface waves [30]. Specular reflection waves are the first component seen in a response, followed by the Rayleigh surface waves. The whispering gallery waves are a smaller contribution. A shell response consists of specular reflections and Lamb surface waves [31]. A Lamb wave propagates around the outside of a shell.

These responses are described by the partial-wave series and have been studied in depth using Sommerfeld–Watson transformation [30,32] and resonant scattering theory (RST) [33,34]. The data used throughout this research have been simulated for a monostatic plane wave incident to elastic spheres and shells using RST. The backscattered response for a solid sphere was generated as described in [33], and the backscattered response for a shell was generated as described in [34].

Backscattered responses were simulated using the following materials: aluminum, stainless steel 347, tungsten carbide, and granite. The longitudinal and shear speed of sound in the materials examined throughout this research were taken from the Engineering Toolbox [35] and are reproduced in Table 1. In practice, typically a probe pulse such as a linear frequency-modulated (LFM) waveform is transmitted to the target of interest.

The measured scattered response is then match filtered. A deconvolution may be performed, but the match-filtering operation itself is approximately equivalent to a finite band deconvolution.

Table 1. Material properties used for simulating backscattered responses. Data from the Engineering Toolbox [35].

Material	Density (kg/m^3)	Longitudinal (m/s)	Shear (m/s)
Aluminum	2712	6420	3040
Granite	2700	4500	3500
Stainless Steel 347	7900	5790	3100
Tungsten Carbide	13,800	6860	4185

A sampled version of the continuous-time backscattered response was simulated. The solution was sampled based on the dimensionless frequency ka. There were 4096 dimensionless frequencies linearly spaced between the real frequency values of 500 Hz and 15.6 kHz for each material. The RST solution resulted in the continuous time solution. Two different fluids filled the shells: air and oil octane. The corresponding material properties were reproduced from the Engineering Toolbox in Table 2.

Table 2. Interior material properties used for simulating backscattered responses. Data from the Engineering Toolbox [35].

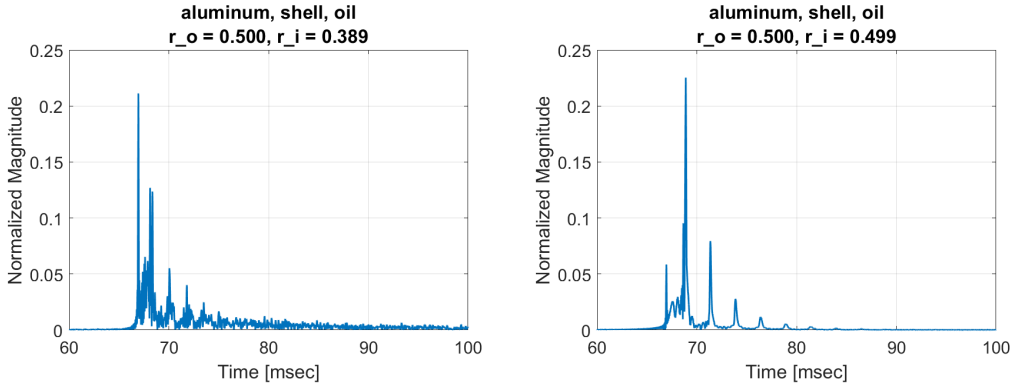
Material	Density (kg/m^3)	Speed of Sound (m/s)
Air	1.2	343
Octane	702	1171

The thickness of the shell greatly impacts the amount of resonance in the response. This parameter is quantified by $h = (r_o - r_i)/r_i$, where r_o is the outer radius and r_i is the inner radius. An example of this impact is shown in Figure 1, where Figure 1a,b shows a thick ($h > 0.1$) and thin ($h < 0.01$) shell, respectively. There are clear and distinct resonances in the thin shell compared with the thick shell. In Figure 1, the specular reflection is the main component seen in the thick shell, while resonance is the main component in the thin shell. The shell's outside radii were set to 0.5 m, while the shell thickness was set to 50 linearly spaced values between 0.001 and 0.901. The thickness of the shell was varied for 200 linearly spaced thickness parameter $h = (r_o - r_i)/r_i$ values within the range h = [0.001–0.901]. The radii on the solid spheres were 50 linearly spaced values between 0.4995 m and 0.0495 m, This is denoted in Table 3.

Table 3. Target properties used for simulating backscattered responses.

Geometry	Interior	Radius	Thickness	Responses Generated
Shell	Air, Octane Oil Aluminum, Granite, Stainless Steel, or Tungsten Carbide	0.5 m	0.001–0.901	400
Sphere		0.4995–0.0495 m	-	200

Independent and identically distributed (i.i.d.) complex Gaussian noise was added to the simulated backscattered signals. Noise was added for two different signal-to-noise-ratio (SNR) cases: 5 dB and 20 dB. The definition of the SNR in this work is $SNR = 10\log_{10}(P_s/\sigma)$ for the signal power P_s and Gaussian standard deviation σ. A higher SNR will result in increased classification due to less signal contamination. After the inclusion of additive noise, the backscattered responses were normalized to the unit of energy.



(**a**) Simulated thick shell response, $h = 0.499$ (**b**) Simulated thin shell response, $h = 0.001$

Figure 1. The normalized magnitude of simulated (**a**) thick and (**b**) thin backscattered responses for an aluminum shell filled with octane. The specular reflection is the main component seen in the (**a**) thick shell, while resonance is the main component in the (**b**) thin shell. The signals have been normalized to the unit of energy, and responses have been zoomed in to the area of interest. There was no noise added to these signals.

2.2. Signal Segmentation

The starting and stopping indices of the signal were estimated using an alternating hypothesis: Page's test [36]. The estimator was based on the cumulative sum of the logarithm of the ratio of the probability distribution function (pdf) of the signal present to the noise pdf. A brief description is presented here. Estimation was performed for the magnitude of the data squared. In the case of the noise-only signal, this becomes chi-squared distributed as the sum of the squared i.i.d. standard normal random variables ($N_R = N_I \sim \mathcal{N}(0,1)$) is a chi-squared distribution with two degrees of freedom:

$$Y = |N|^2 = N_R^2 + N_I^2 \sim \chi_2^2, \tag{1}$$

The distribution takes the reduced form of

$$f_0(x) = \frac{1}{2} e^{-x/2}. \tag{2}$$

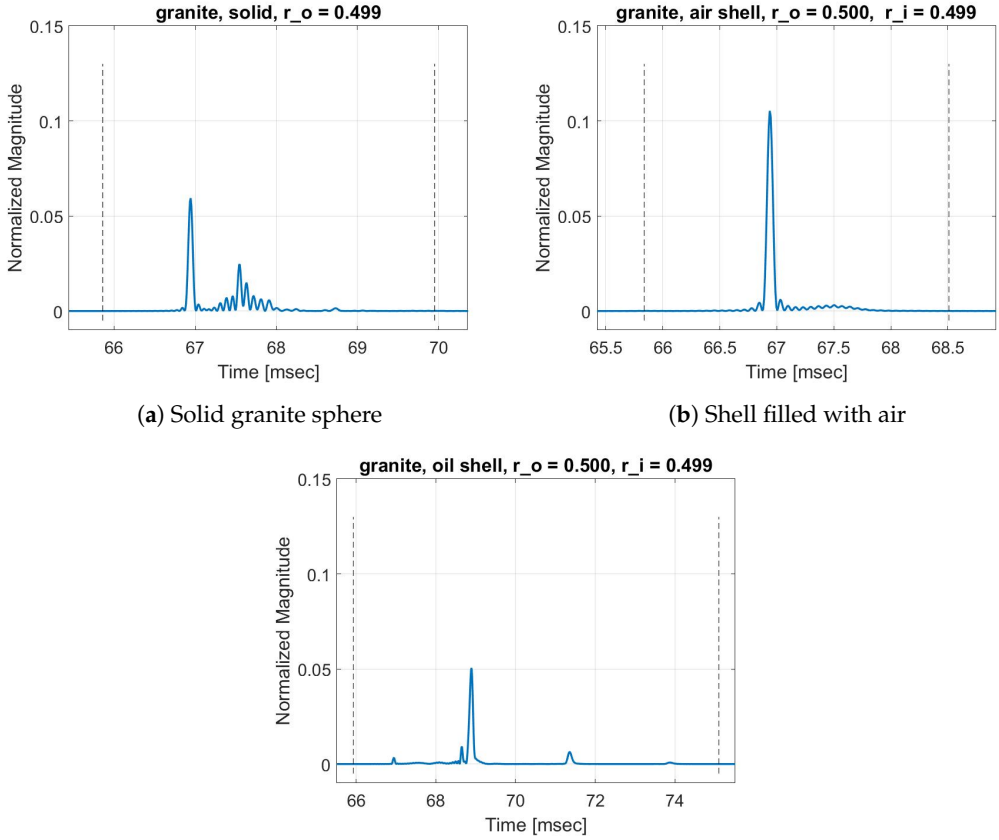
When the signal is present, the distribution can be described by an i.i.d. complex Gaussian with constant means and variances, where $S_R = S_I \sim \mathcal{N}(\mu, \sigma^2)$. The random variable will have a non-central chi-squared distribution with two degrees of freedom and a non-centrality parameter $\delta = 2(\mu/\sigma)^2$. The density function can be written as

$$f(x;\delta) = \sum_{k=0}^{\infty} \frac{\delta^k e^{-\delta/2} x^k e^{-x/2}}{(k!)^2 2^{2k+1}}. \tag{3}$$

A locally optimal detector was employed as an approximation of the log-likelihood ratio due to the infinite summation. This is a linear function of the data, as only the first two terms of the summation are nonzero [36]. A bias was determined by using Dyson's method [36]. The signal was segmented using a cumulative summation of the data, which takes the iterative form of

$$T_k = \max[0, T_{k-1} + g(x)] \tag{4}$$

where $T_0 = 0$ and $g(x)$ is the locally optimal detector with bias. When T_k is greater than some threshold, the signal has been detected. A similar method can be used to find the end of the signal. Due to the cumulative summation, there is a delay associated with the start of the signal. To mitigate loss of the start of the signal, an empirically chosen 1 ms value was selected for the start of the signal index. Figure 2 shows an example of the segmented signals for the granite solid, shell filled with air, and shell filled with oil.



(**a**) Solid granite sphere

(**b**) Shell filled with air

(**c**) Shell filled with octane oil

Figure 2. Simulated backscattered responses' start and stop signal indices for the granite (**a**) sphere with a radius of 0.499 m, (**b**) shell filled with air, and (**c**) shell filled with octane oil, with an SNR level of 5 dB. The shells had an outside radius of 0.5 m and inside radius of 0.499 m. The simulated signal is shown in solid blue, and the starting and stopping indices are shown in black dashed lines. Signals have been normalized to the unit of energy.

2.3. Time-Frequency Representations

After the data generation, the backscattered responses of the targets were projected into time-frequency representations. Two time-frequency representations were employed: spectrograms generated from STFTs and scalograms generated from the CWT. The STFT and CWT representations were chosen, as they are two standard methods commonly used. The continuous-time STFT is described below to demonstrate the similarities to the continuous wavelet transform. The STFT of a signal $x(t)$ is defined as follows:

$$STFT(\tau, \omega) = \int_{-\infty}^{+\infty} x(t) w(t-\tau) e^{-j\omega t} dt \tag{5}$$

where τ and ω correspond to the time and frequency, respectively, and $w(t)$ is a windowing function used to reduce spectral leakage. The Hamming window function of length of 50 samples was used in this application. A 90% overlap was employed with an FFT equivalent to the next power of two of the length of the signal. Typically, this was a 2048 point FFT. The spectrogram is the squared magnitude of the STFT, or

$$S(\tau, \omega) = |STFT(\tau, \omega)|^2. \tag{6}$$

The spectrograms of the segmented backscattered responses generated from stainless steel 347 with a 20 dB SNR are displayed in Figure 3. The spectrograms are depicted for a shell filled with air in Figure 3a, a shell filled with oil in Figure 3b—both having an outside

radius of 0.50 m and inside radius of 0.49 m—and a solid sphere with a radius of 0.49 m in Figure 3c. Figure 3d shows the spectrogram for a shell with an inner radius of 0.343 m filled with air. This inset was included to demonstrate the morphing of resonances as a function of the target size. Notice how the resonances varied depending on the target geometry (shell or solid), the inside material (air, octane oil, or solid), and the target size. For example, there were distinct and localized resonances between 0 and 10 kHz for the shell filled with air (Figure 3a), while there were complex resonances across the simulated frequency band for the shell filled with oil (Figure 3b). When visually comparing a shell filled with air at different thicknesses (Figure 3a,d), the resonances spread across the simulated frequency band when the shell was thick. The thick shell spectra (Figure 3d) was most similar to the solid sphere (Figure 3c), consisting of a strong specular reflection and wide resonances in this case.

The CWT for the scale parameter $a > 0$ and translation parameter b is

$$CWT(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} x(t)\psi^* \left(\frac{t-b}{a} \right) dt. \tag{7}$$

where the previously mentioned notation holds and $\psi(\cdot)$ is the wavelet that can contract and dilate, allowing for a variable time-frequency resolution. The Morlet wavelet was employed in this research and can be described as

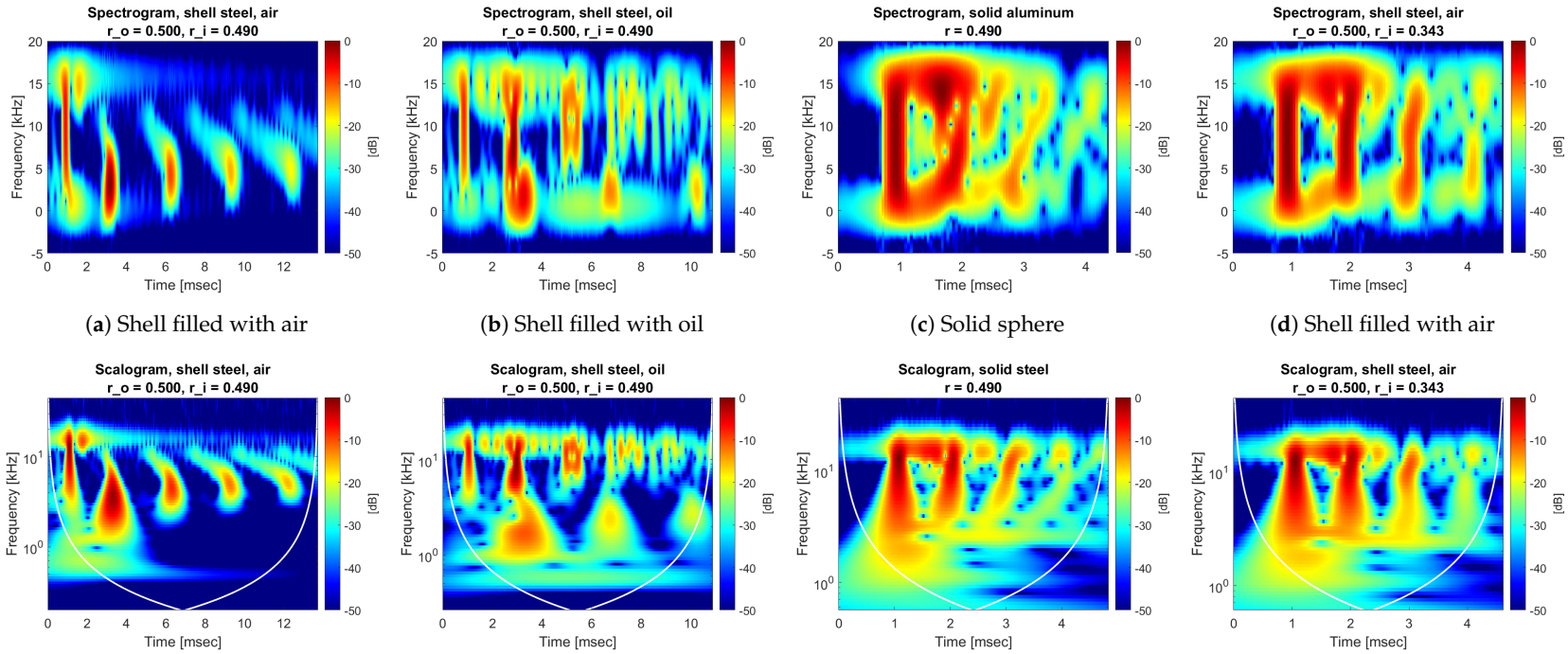
$$\psi_0(t) = \pi^{-1/4} e^{j\omega_0 t} e^{-t^2/2} \tag{8}$$

for a frequency $\omega_0 = 2\pi f_0$ and time t. The Morlet wavelet belongs to the class of analytic wavelets due to its complex oscillation. This wavelet is sometimes referred to as a Gabor wavelet and is a Gaussian-modulated with a plane wave [37]. The oscillation provides frequency localization, while the Gaussian envelope provides time localization [38]. The scalogram is the magnitude squared of the CWT:

$$C(a,b) = |CWT(a,b)|^2. \tag{9}$$

The scalograms of the segmented, backscattered responses generated from stainless steel 347 with a 20 dB SNR are displayed in Figure 3e–h. Scalograms are depicted for a shell filled with air in Figure 3e, a shell filled with oil in Figure 3f—both having an outside radius of 0.50 m and inside radius of 0.49 m—and a solid sphere with a radius of 0.49 m in Figure 3g. Figure 3h shows the scalograms for a shell with an inner radius of 0.343 m filled with air. This inset was included to demonstrate the morphing of resonances as a function of the target size. Notice how the resonances varied depending on the target geometry (shell or solid), the inside material (air, octane oil, or solid), and the target size. Similar visual patterns described when examining the spectrogram representations held in describing the scalogram representations. The white lines on the scalograms indicate the cone of influence (COI). Features present outside of the COI should be questioned, as they may be caused by edge effect artifacts and boundary effects due to the wavelet being stretched outside the observation interval.

The constant window size force spectrograms have a fixed resolution and may not accurately capture information on non-stationary signals. The scalograms had variable resolutions due to the scaling and translation parameter. This manifested as high-frequency components having good time resolutions but poor frequency resolutions and vice versa. This trade-off can be seen in Figure 3e–h, where the low-frequency component is spread over time and high-frequency component is localized at a time. Typically, scalograms are a more accurate representations of non-stationary signals.



(**a**) Shell filled with air (**b**) Shell filled with oil (**c**) Solid sphere (**d**) Shell filled with air

(**e**) Shell filled with air (**f**) Shell filled with oil (**g**) Solid sphere (**h**) Shell filled with air

Figure 3. Spectrograms (**a**–**d**) and scalograms (**e**–**h**) of the segmented backscattered responses generated from stainless steel 347 with a 20 dB SNR. The white lines on the scalograms represent the cone of influence. Spectra are depicted for a shell filled with air ((**a**) spectrogram and (**e**) scalogram), a shell filled with oil ((**b**) spectrogram and (**f**) scalogram), and a solid sphere with a radius of 0.49 m ((**c**) spectrogram and (**g**) scalogram). All shells had an outside radius of 0.50 m and inside radius of 0.49 m. The inset (**d**) depicts the spectrogram, and the other inset (**h**) depicts the scalogram of a shell filled with air with an inside radius of 0.343 m. Notice how the resonances vary depending on the geometry (shell vs. solid) and inside fluid used for the simulation (air vs. octane oil). The resonances also morph as a function of the target size, as can be seen when comparing insets (**a**–**d**) and (**e**–**h**). The unit is decibels and in reference to the highest pixel intensity.

2.4. Multi-Class Classification

A convolutional neural network (CNN) was used for classification. Other deeper network architectures, such as YOLO, ResNet, and iterations of them, have been used for similar tasks (as described in [8,39,40]). However, satisfactory classification results were achieved using the following network structure while explaining the network choices and relating the interpretation into the physical domain. The filter size was set to 5×5. Batch normalization was used after each two-dimensional convolution, and a max pooling layer with a ReLU activation function followed [41]. The max pooling decreased the output size by a factor of two. After the last ReLU activation was a max pooling layer followed by a dense layer with 128 neurons. Lastly, a fully connected layer with the number of neurons equal to the number of classes and a softmax activation were included.

All data were resized to 128×128 via bicubic interpolation along with a low-pass anti-aliasing filter prior to training and testing using MatLab's imresize function [29]. The data were split as 80% for training and validation and 20% for testing. A common random seed was used. The networks were trained using the Adam optimizer with the default parameters [42]. Early stopping was employed to combat overfitting [43]. The early stopping strategy employed in this work was patience-based for the validation loss; that is, if the validation loss increased across 10 training cycles, then training terminated. The validation loss was calculated for every other training cycle. The cross-entropy loss function was minimized. The multi-class cross-entropy loss function for the true label y and predicted output $\hat{y}_n^c$ at observation n for class c is defined as follows:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} (y_n^c \ln \hat{y}_n^i) + (1 - y_n^c) \ln(1 - \hat{y}_n^c). \tag{10}$$

The number of convolutional layers, number of filters, and the initial learning rate were determined through Bayesian optimization by assuming a prior Gaussian process [44]. The acquisition function employed was expected to show improvement and iterated 15 times [44]. The 80% subset of training data was split into 5 cross-validation folds. Each iteration of the Bayesian optimization was performed on the five folds. The average loss across the five folds was minimized. This allowed for the network topology to be optimized for the classification task while ensuring generalizability. More information on the Bayesian optimization method employed in this research is included in Appendix A. Bayesian optimization was chosen over random searches for hyperparameter tuning to minimize the network tuning time [45]. Figure 4 depicts an example of the network structure, where N_f and N_l are the number of filters and the number of layers determined through Bayesian optimization, respectively.

Networks with the goal of classifying material types (aluminum, tungsten carbide, stainless steel 347, or granite), target geometries (solid or shell), and the interior fluid of the target (solid, air, or octane oil) were trained and tested. This was carried out to determine which components of the representations were most influential in the classifier decisions and to uncover any relations between the network structure and classification task. Tables 4–6 show the hyperparameter search space and the optimizer results for the networks. The hyperparameters reported in Tables 4–6 were used to generate the results discussed.

Upon examination of the selected hyperparameters in Tables 4–6, the networks trained on scalograms typically required fewer layers and filters. This may be due to the scalogram representation having better discriminatory features when compared with the spectrogram representation. The spectrogram networks have increased capacity, as the network requires additional parameters in order to decode the spectra. A similar remark can be made for the networks with regard to the SNR, as a lower SNR requires typically requires increased network capacity. This is due to the network needing to filter out the noise prior to finding discriminatory features.

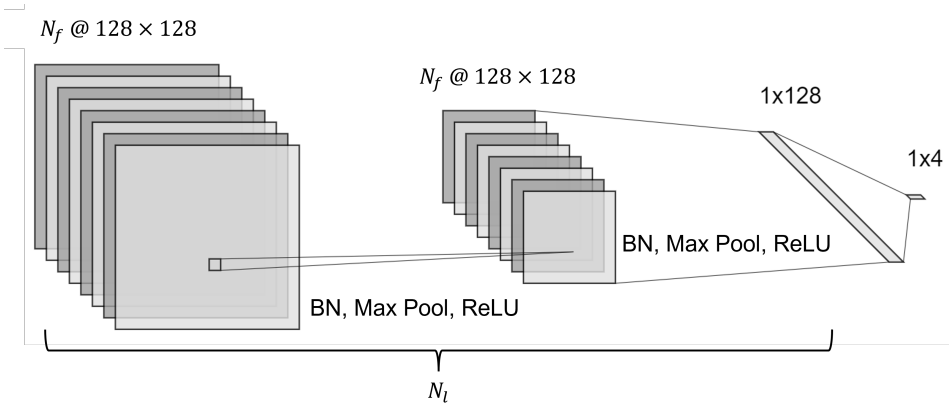


Figure 4. A network structure example. The number of filters N_f and the number of convolutional layers N_l were determined through Bayesian optimization. Each convolutional layer was followed by batch normalization, max pooling, and an ReLU activation function. The second-to-last layer was a fully connected layer with 128 neurons. The last layer was a fully connected layer with a number of neurons equal to the number of classes (four in this depiction).

Table 4. Number of convolution layers and filters and the initial learning rates (LRs) for the CNN chosen through Bayesian optimization for classification of the geometry (sphere or shell). The hyperparameter search space is also reported.

	Spectrogram			Scalogram		
SNR	**Layers**	**Filters**	**LR**	**Layers**	**Filters**	**LR**
Search Space	[1, 4]	[2, 9]	$[1 \times 10^{-5}, 1 \times 10^{-1}]$	[1, 4]	[2, 9]	$[1 \times 10^{-5}, 1 \times 10^{-1}]$
5 dB	2	4	2.48×10^{-4}	1	6	7.36×10^{-4}
20 dB	4	9	2.01×10^{-3}	1	4	4.24×10^{-4}

Table 5. Number of convolution layers and filters and the initial learning rates (LRs) for the CNN chosen through Bayesian optimization for classification of the interior filling (solid, air, or oil octane). The hyperparameter search space is also reported.

	Spectrogram			Scalogram		
SNR	**Layers**	**Filters**	**LR**	**Layers**	**Filters**	**LR**
Search Space	[1, 4]	[2, 9]	$[1 \times 10^{-5}, 1 \times 10^{-1}]$	[1, 4]	[2, 9]	$[1 \times 10^{-5}, 1 \times 10^{-1}]$
5 dB	4	5	5.24×10^{-3}	2	3	7.68×10^{-3}
20 dB	1	9	6.51×10^{-4}	1	3	9.80×10^{-4}

Table 6. Number of convolution layers and filters and the initial learning rates (LRs) for the CNN chosen through Bayesian optimization for classification of the material (aluminum, granite, stainless steel 347, or tungsten carbide). The hyperparameter search space is also reported.

	Spectrogram			Scalogram		
SNR	**Layers**	**Filters**	**LR**	**Layers**	**Filters**	**LR**
Search Space	[1, 4]	[2, 9]	$[1 \times 10^{-5}, 1 \times 10^{-1}]$	[1, 4]	[2, 9]	$[1 \times 10^{-5}, 1 \times 10^{-1}]$
5 dB	4	9	9.61×10^{-4}	2	4	6.67×10^{-4}
20 dB	2	6	1.13×10^{-4}	2	8	9.37×10^{-4}

Computations were performed in Matlab 2022a v9.12.0 with the Deep Learning Toolbox [29] using an Intel(R) Core(TM) i9-10900K CPU with 32 GB of RAM and an NVIDIA Quadro P1000 GPU. There were 15 iterations of the optimization process, resulting in a total of 75 trained networks (across 5 folds) per hyperparameter search. Each optimization took roughly 35 min to perform, and these are reported in Tables 7 and 8 for the 5 dB and 20 dB SNRs, respectively. While the optimization times do not facilitate real-time applications, the parameters reported within Tables 4–6 may aid future researchers by reducing their hyperparameter search space.

Table 7. Hyperparameter optimization time for networks trained on data with 5 dB SNR.

Classification Task	Spectrogram Time (min:s)	Scalogram Time (min:s)
Geometry	38:11	35:47
Interior material	33:11	35:05
Material type	35:44	34:00

Table 8. Hyperparameter optimization time for networks trained on data with 20 dB SNR.

Classification Task	Spectrogram Time (min:s)	Scalogram Time (min:s)
Geometry	36:10	33:12
Interior material	35:41	32:29
Material type	42:05	32:39

2.5. Performance Metrics

The overall accuracy (OA) can be used to determine on average how well a classifier is performing. It is defined as the number of correctly classified responses divided by the total number of attempted classified responses and multiplied by 100%:

$$\text{OA} = \frac{\#\,\text{correctly classified}}{\#\,\text{attempted classified}} \cdot 100\%. \tag{11}$$

Confusion matrices were used to see the incorrectly classified predictions for each class. On one axis is the true label, while the other has the predicted label. The values indicate the amount of responses classified at that specific instance. A classifier that is 100% accurate will have values only on the diagonal. The confusion matrices in this work were row-normalized so that the value depicted was a percentage of the true class label.

2.6. Gradient-Weighted Class Activation Mapping (Grad-CAM)

Selvaraju et al. developed an explainable technique: gradient-weighted class activation mapping (Grad-CAM) [25]. Other XAI techniques, such as local interpretable model-agnostic explanation (LIME) [46] or CNN feature map examinations [11], have been employed to explain sonar classification choices. This work takes a similar approach in the explanation but uses a model-specific technique to relate the classifier choices to the physical domain. Grad-CAM falls in the general class of gradient-based techniques and uses backpropagation on trained networks to explain network choices. Grad-CAM is a generalization of class activation mapping (CAM), described in [47], and does not require a specific network structure.

Following the notation in [25], Grad-CAM calculates an input relevance score by first sending a test image through the trained network, resulting in an output score y^c for class c. The gradient of the output score $\partial y^c / \partial A^k$, with respect to the kth feature map activation A^k of a specified convolutional layer, is then computed. Typically, the last convolutional layer is chosen. The gradients are then global average-pooled across the channel depth,

essentially averaging all pixels in the feature map. These averaged gradients α_k^c can be thought of as weights

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{12}$$

for the width and height indices i and j, respectively, with Z total pixels. The global average-pooling is apparent through the two summations and the division of Z. The averaged gradient weights are then multiplied by the corresponding feature activation maps. The ReLU activation is used for rectification and ensures only positive influences are considered [25]. The output of the ReLU function is of the same size as the convolutional layer and is upsampled to the input image size.

Intuitively, when a gradient is large, the output score depends on this location. This is essentially what Grad-CAM is depicting. Visualization is performed by using a heat map to highlight class-specific features and provide a relevance value for the input pixels.

3. Results

The overall classification results for the material type, geometry, and interior fill are reported for the different SNRs. Next, the interesting confusion matrices are examined. Lastly, the trained classifiers are examined, using Grad-CAM to explain the network choices and relate the informative features to the physical domain.

3.1. Overall Accuracy

The overall accuracy and standard deviation for the CNNs trained on the simulated data to classify target geometry are in Table 9, while those for the interior material are shown in Table 10, and those for the material type are in Table 11. The reported results were generated by training a network using the hyperparameters listed in Tables 4–6 on the training and validation data used for the cross-validation Bayesian optimization. Testing was performed using the disjointed 20% data subset. The reported results are from across 10 random seeds to determine a statistical description of the network.

Table 9. The overall accuracy and standard deviation for classification of geometry of the target (solid or shell). The largest average overall accuracy (OA) and corresponding standard deviation (Std. Dev.) are denoted in bold.

	Spectrogram		Scalogram	
SNR | Metric	**OA (%)**	**Std. Dev. (%)**	**OA (%)**	**Std. Dev. (%)**
5 dB	96.83	1.34	97.75	1.19
20 dB	97.75	0.68	**97.91**	**1.93**

Table 10. The overall accuracy and standard deviation for classification of the inside of the target. The largest average overall accuracy (OA) and corresponding standard deviation (Std. Dev.) are denoted in bold.

	Spectrogram		Scalogram	
SNR | Metric	**OA (%)**	**Std. Dev. (%)**	**OA (%)**	**Std. Dev. (%)**
5 dB	78.50	2.82	76.67	2.18
20 dB	**78.58**	**2.18**	78.16	1.29

Typically, a higher SNR results in a higher average classification accuracy and lower standard deviation. The scalograms tend to have a higher average classification accuracy. However, there was no discernible trend in the corresponding standard deviations. Both trends were expected, as the increased SNR provided a cleaner representation for the network to classify, and the scalogram representation provided increased localization of the robust features due to the contraction and dilation of the wavelets. There was a

negligible difference between the two SNR classification accuracies, accounting for the standard deviation of the random seeds. This demonstrates the CNNs' ability to perform noise suppression across a difference of 15 dB. In future work, this can be taken to the extreme, and analysis across different SNR levels can be used to determine the failure mode of the classifier. Given these results, and by leveraging the knowledge learned in the hyperparameter optimization, while scalograms provide negligible increased classification accuracy compared with their spectrogram counterpart, the scalogram networks typically have fewer layers and filters, which decreases the number of learned parameters, the complexity of the network, and the network training time.

Table 11. The overall accuracy and standard deviation for classification of material type. The largest average overall accuracy (OA) and corresponding standard deviation (Std. Dev.) are denoted in bold.

	Spectrogram		Scalogram	
SNR | Metric	**OA (%)**	**Std. Dev. (%)**	**OA (%)**	**Std. Dev. (%)**
5 dB	83.50	1.79	89.25	2.40
20 dB	83.50	1.91	**90.91**	**1.32**

3.2. Confusion Matrices

The confusion matrices for the highest average accuracies reported Tables 9–11 are shown in Figures 5–7. The confusion matrices list the classification results across the 10 random seeds. The units of the confusion matrices have been row-normalized to easily determine a percentage per class of correct or incorrect classification. For example, in Figure 5, 3.3% of the shells were misclassified as solid spheres.

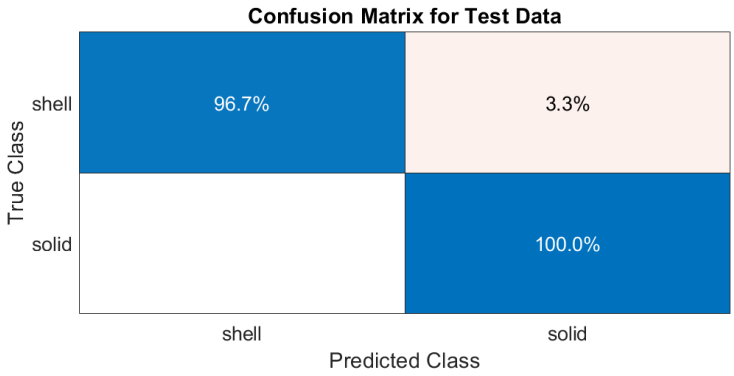


Figure 5. Confusion matrix for the CNN trained to classify geometry (shell or solid) with the highest classification accuracy (Table 9). The network was trained using the scalogram representation with a 20 dB SNR, resulting in a 97.91% average overall accuracy. This network consisted of one convolutional layer with four feature maps per layer (Table 4). The blue boxes indicate correct classification while the orange boxes indicate incorrect classification.

The highest classification accuracy occurred when the network was instructed to train based off of the target geometry (solid or shell). The lowest accuracy occurred when the network was instructed to discriminate between the interior materials (air, oil, or solid). These trends were a result of the simulated shells having similar acoustic spectra, regardless of the internal fluid, when the shells were thick. We recall that the radii of the shells were randomly split across the training and testing dataset, as described in Section 2.4. These classification results represent an average across the randomly separated radii. However, the classification result may still be explained through examination of the different feature representations. An example of this similar response can be seen in Figure 8, where the spectrograms were generated with a thick shell ($h \sim 0.9$) made from tungsten carbide. Figure 8a was generated by using air as the internal fluid, while Figure 8b used octane oil. There is little visual difference between the two spectra, which in turn

confused the classifier. When the shells had a reduced thickness, as seen when comparing the images in Figure 3a,b, the classifier was able to distinguish between the interior fluid. The thickness where the visually similar acoustic spectra began to differ occurred sooner in the scalogram representation. This gave the network more variability in the spectra providing the increased accuracy when the network was trained on the scalograms. A classifier trained to discriminate between a solid and shell will automatically group the air- and oil octane-filled shells in a class, resulting in the increased classification accuracy seen in Table 9.

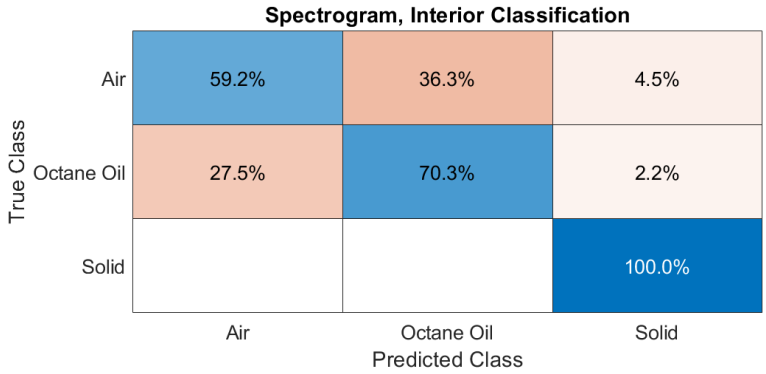


Figure 6. Confusion matrix for the CNN trained to classify interior material (air, octane oil, or solid) with the highest classification accuracy. The network was trained using the spectrogram representation with a 20 dB SNR, resulting in a 78.58% average overall accuracy (Table 10). This network consisted of one convolutional layer with nine feature maps per layer (Table 5). The blue boxes indicate correct classification while the orange boxes indicate incorrect classification.

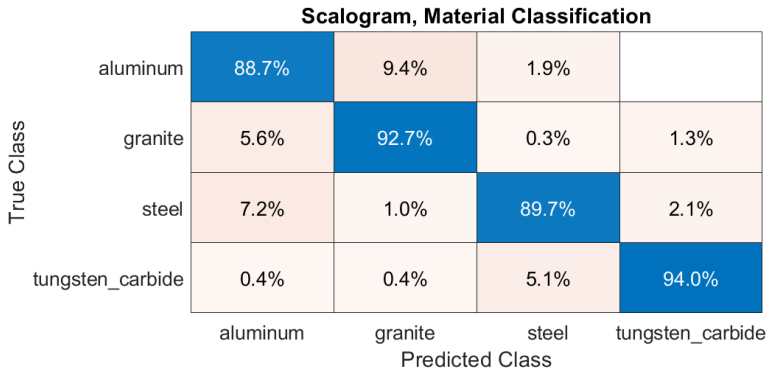
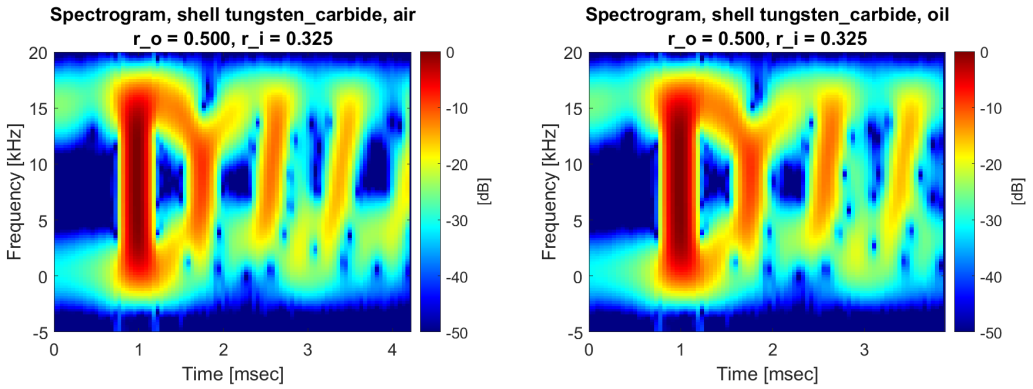


Figure 7. Confusion matrix for the CNN trained to classify material (aluminum, granite, stainless steel 347, or tungsten carbide) with the highest classification accuracy. The network was trained using the scalogram representation with a 20 dB SNR, resulting in a 90.91% average overall accuracy (Table 11). This network consisted of two convolutional layers with eight feature maps per layer (Table 6). The blue boxes indicate correct classification while the orange boxes indicate incorrect classification.

3.3. Grad-CAM

Grad-CAM was used to determine what the CNN was selecting as the most influential features for each classification task. The Grad-CAM results are presented as heat maps that show the spatial locations of large gradients. The axes are in units of pixels and representative of the time and frequency axes. The color bar was normalized between 0 and 1. The heat maps are accompanied by the corresponding input image. Note that the input images are not in a decibel scale, as the networks were trained using normalized linear units. The advantage to using Grad-CAM is the location of the class-specific influence on the spectra can be found. These are the locations of the largest gradients. This can

provide insight on feature extraction algorithms (i.e., intuitively bias the classifier by forcing it to focus on informative features) and aid in classifier debugging. Examination of the Grad-CAM heat maps on the simulated data allows relations to be drawn between the important features and physical scattering mechanisms.



(**a**) Thick shell filled with air (**b**) Thick shell filled with oil

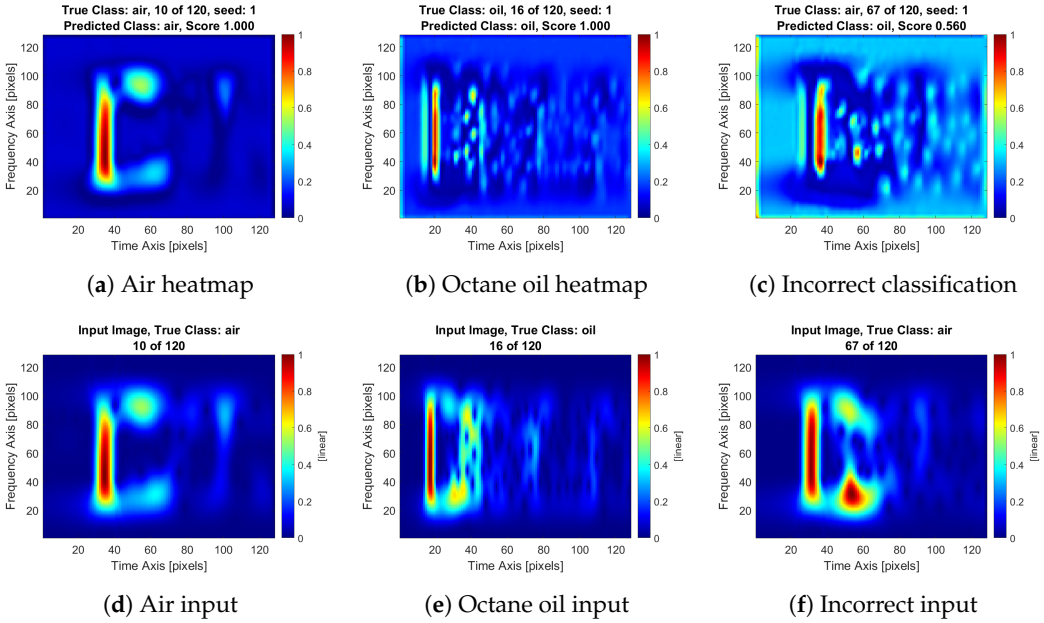
Figure 8. Similar spectrograms generated from tungsten carbide with a 20 dB SNR for thick shells filled with (**a**) air and (**b**) octane oil. The similar spectra results in the interior classification network had decreased accuracy. Units are in decibels and are in reference to the largest pixel intensity.

Grad-CAM was performed on the CNNs for the reported confusion matrices. The resulting heat maps were visually similar to the input images for the classification of the interior material and target geometry. This was due to the initial convolution layers detecting semantically meaningful objects, and both networks were one convolutional layer deep. An example of these heat maps and the corresponding test images are shown in Figure 9. The heat maps depicted are for the correctly classified air-filled shell (Figure 9a), octane oil-filled shell (Figure 9b), and an air-filled shell that was incorrectly classified as an octane oil-filled shell (Figure 9c). The corresponding input test images are depicted in Figure 9d–f, respectively. The corresponding classification scores are listed in the title of the Grad-CAM heat maps.

The CNN depends on the first feature component, typically due to specular or geometric scattering, when choosing the air class and higher-order, more complex resonances when choosing the octane oil-filled shell class. This is evident in the comparison of the heat maps in Figure 9a,b. Insight into the CNN model's failure in misclassification can be gained through examining Figure 9c. Figure 9d–f depict the corresponding input test image. This test image was incorrectly classified as an octane oil-filled shell and had a classification score of 0.560, which was split between the two classes. The CNN found a similar oil structure within the input image but also relied on features external to the resonances, as is evident from the highlighted features on the exterior of the image. To mitigate this failure mode of the CNN, a segmentation processing step could be included in the classification pipeline. This would likely increase the classification accuracy and decrease the training time, since the network mainly relies on semantic clues within the image.

Figure 10 depicts the heat map for the CNN trained on spectrograms to recognize the target geometry. The input test image was the same as the octane oil input image in Figure 9e. A comparison between Figures 9b and 10 provides insight into the features that are important to different classification tasks. The classifier used to discriminate between interior materials depends on higher-order, more complex resonances (Rayleigh or Lamb surface waves), while the classifier trained to discriminate between the target geometries relies on the shape of the specular scattering and first resonance. This can be leveraged when designing a classification pipeline, as the classification task must be taken into account. If the target geometry is being classified, then the feature representation can focus on the start of the signal and the specular reflection rather than its entirety. This provides an

automatic dimensional reduction in the signal truncation, resulting in decreased classifier complexity and training time.



(**a**) Air heatmap (**b**) Octane oil heatmap (**c**) Incorrect classification

(**d**) Air input (**e**) Octane oil input (**f**) Incorrect input

Figure 9. Grad-CAM heat maps for the CNN trained to classify the interior material. Correct classification results are shown for the (**a**) air and (**b**) octane oil interior fillings. An incorrect classification heat map (**c**) where an air-filled shell was incorrectly classified as an octane oil-filled shell is also shown. Insets (**d**–**f**) depict the input test image. All units are linear and have been normalized between 0 and 1.

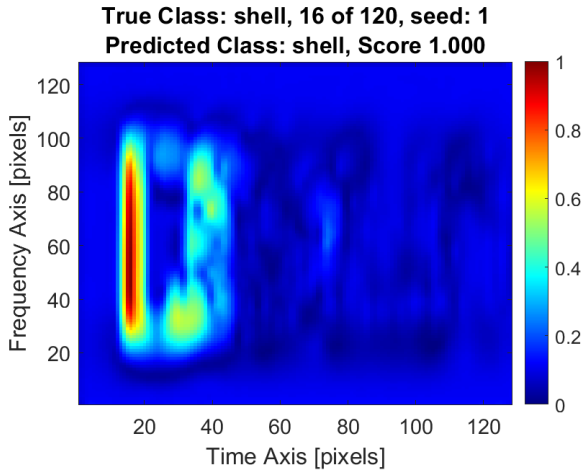
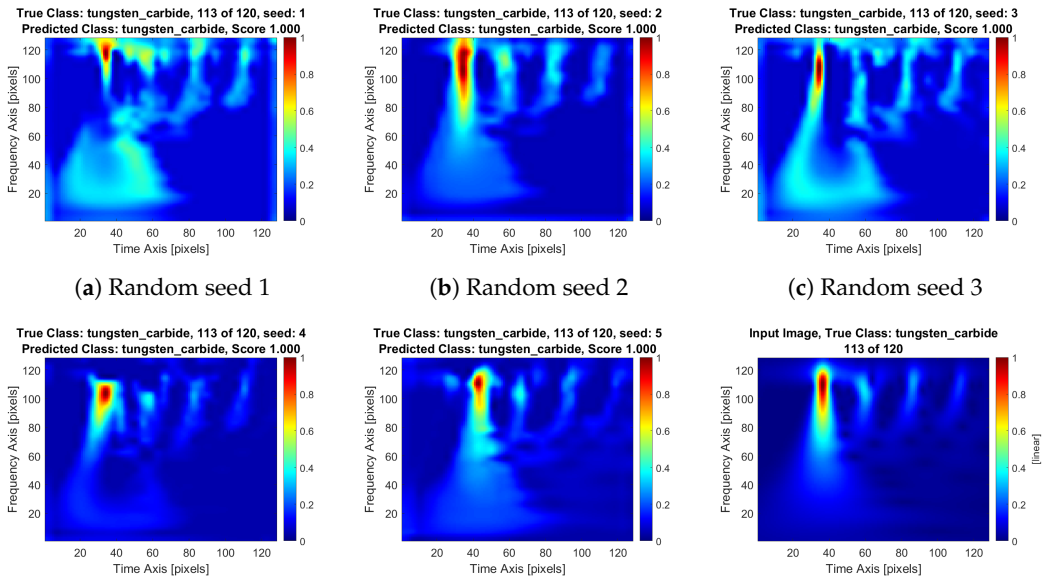


Figure 10. Grad-CAM heat map for the CNN trained on spectrograms to discriminate between shell and solid, generated from the same test image in Figure 9e.

To determine how the varying random seed impacted the classification accuracy, Grad-CAM was performed on the CNNs trained on the scalograms with a 20 dB SNR to classify the material type. The resulting heat maps for the first five random seeds and the test image are shown in Figure 11. Figure 11a–e depicts the Grad-CAM heat maps for various random seeds, and Figure 11f depicts the corresponding test image. The remaining five seeds were omitted due to space constraints. We recall that this network consisted of two convolutional layers with eight feature filters, meaning the network still relied on semantically meaningful objects. In all cases, the CNN depended on the tungsten carbide specular reflection for discrimination. The CNN was additionally using the resonances

caused by Rayleigh or Lamb surface waves, but they were not as influential. In some cases, the CNN relied on additional low-frequency features (Figure 11a,c), while others depended on localized resonances (Figure 11b,d,e). An additional processing step could be included in the classification pipeline to highlight the high-frequency resonances and force the CNN to depend on these features, which are known from this analysis to be consistent across random seeds and impact classification.

Throughout examination of the Grad-CAM results, in order to highlight features that were determined to be relevant, various forms of augmentation could be used to increase the dataset and robustness of the network. Additionally, the insight gained from the heat maps allow recommendations to be made when designing different classifiers. For example, if the target geometry is to be classified, then the majority of the information relevant to the network is within the specular scattering, as seen in Figure 10. This places less relevancy on the end of the return, and the signal detector can be adjusted to focus on the initial return. This would automatically decrease the input dimensionality and focus the network beforehand on the relevant features, thereby intuitively positively biasing the results and decreasing the training time.



(**a**) Random seed 1 (**b**) Random seed 2 (**c**) Random seed 3

(**d**) Random seed 4 (**e**) Random seed 5 (**f**) Input image

Figure 11. Grad-CAM heat maps for the CNN trained to classify the target material initialized using (**a**–**e**) random seeds 1–5, respectively. All seeds resulted in the correct classification of the (**f**) tungsten carbide input image. The classification scores are listed in the titles. All units are linear and have been normalized between 0 and 1.

4. Conclusions

Simulated backscattered responses of various materials, shapes, and sizes were generated prior to projection onto two time-frequency representations: spectrograms generated using the short time Fourier transform and scalograms generated using the continuous wavelet transform. Multiple convolutional neural networks were trained to classify the material type, target geometry, and interior material of the target. Bayesian optimization was used to determine the number of layers, number of feature maps per layer, and the initial learning rate. This resulted in classifiers that were optimized for the specified classification task. The trained networks were examined by using an explainable artificial intelligence technique—gradient-weighted class activation mapping (Grad-CAM)—to determine the post-training features used for classification. The Grad-CAM results were depicted using heat maps, representing the spatial locations of large positive gradients.

The scalogram representation provided a negligible increase in the average classification accuracy over the spectrograms. The networks trained to discriminate between target geometries resulted in the highest classification. The networks trained to discriminate the target's interior material had the lowest accuracy. The main feature highlighted when examining the CNNs trained to classify the interior of the material was the specular reflection, with a small portion of the resonances being used for classification. This network contained one convolutional layer. Typically, the initial layers of the network will lock onto spatially important features (such as contours), while deeper layers separate out things that are not visually apparent. The CNN used to classify the material type was two convolutional layers. These CNNs highlighted the resonances for discrimination, but the specular component was also still being used. The analysis performed throughout this investigation can be leveraged when designing classification pipelines by amplifying the meaningful scattering mechanisms and suppressing the less-informative features. Possible classifier failure mode mitigation techniques were discussed, and recommendations for how to intuitively and positively bias classifiers were provided.

Future work can further relate the network-determined features through first simulating the modal rigid and soft residuals. The Bayesian optimization topology can have increased classification to follow the best practices in CNN topology, such as by increasing the number of filters with each additional convolutional layer. Additional complexities can be included in the classification pipeline, such as coating the shell and simulating cylinders and investigating different additive noise models at different SNRs. The latter analysis would further investigate the impact of the SNR on the network analysis and explainability technique. The thickness of the shell and relation to the classification results can be further investigated. Deeper networks can be trained and examined to see if separation between the different scattering mechanisms occurs. Lastly, the analysis of the trained networks can be expanded. The results presented were qualitative through visual examination of the Grad-CAM heat maps. An image similarity measurement, such as the structural similarity index measure (SSIM) [48] or feature similarity index (FSIM) [49], could be used to quantify intra-class features and compare them to inter-class features.

Author Contributions: Conceptualization, A.S.G. and I.K.; methodology, B.K.; software, B.K.; validation, A.S.G., I.K. and B.K.; formal analysis, B.K.; investigation, B.K.; resources, A.S.G. and I.K.; data curation, B.K.; writing—original draft preparation, B.K.; writing—review and editing, A.S.G., I.K. and B.K.; visualization, B.K.; supervision, A.S.G.; project administration, A.S.G.; consulting and mentorship, I.K.; funding acquisition, A.S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Office of Naval Research, Grant Numbers N00014-19-1-2436, N00014-21-1-2420, and the National Defense Science and Engineering Graduate Fellowship Program (2021).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2-D	Two-dimensional
ARD	Automatic relevance determination
Grad-CAM	Gradient-weighted class activation mapping
CM	Confusion matrix
CNN	Convolution neural network
COI	Cone of influence

CWT	Continuous wavelet transform
OA	Overall accuracy
ReLU	Rectified linear unit activation function
RST	Resonant scattering theory
SNR	Signal-to-noise ratio
STFT	Short-time Fourier transform
i.i.d.	Independent and identically distributed

Appendix A. Bayesian Optimization

Bayesian optimization seeks to find a set of globally optimal parameters $\hat{x}$ that minimizes an objective function $f(x)$ across some parameter search space A of x:

$$\hat{x} = \arg\min_{x \in A} f(x). \tag{A1}$$

There are two main components in Bayesian optimization: (1) the surrogate function and (2) the acquisition function. The surrogate function is used as a model of the objective function. The objective function is the CNN. The acquisition function is used to determine which sets of parameters should be sent through the objective function next.

In this case, the surrogate function is modeled using a Gaussian process. A Gaussian process assumes the function can be modeled as a multivariate Gaussian distribution and depends solely on the mean and a covariance function. The covariance function is referred to as the kernel. MatLab's default, the automatic relevance determination (ARD) Matern 5/2 kernel [44], was used. The surrogate function is a probabilistic model of the objective function and is used to determine which parameters will likely result in increased accuracy. It uses Bayes's theorem to update a posterior distribution from a prior distribution by using the scores from the objective function. The score of the objective function is determined by averaging the scores across the five folds. This ensures the generalized hyperparameter selection through cross validation.

The Gaussian process parameters are updated by using the expected improvement plus the acquisition function. The expected improvement is calculated by taking the expectation of the objective function's improvement. There is a trade-off between exploitation and exploration when performing the optimization. Exploitation is driven by the mean of the model and explores new values close to the best results, while exploration is driven by the standard deviation and tries to explore previously unseen regions.

The parameter search spaces specified in Tables 4–6 were uniformly distributed. The learning rate was uniformly distributed over a logarithmic space. After the optimizer is complete, the hyperparameters with the lowest classification score are chosen. A final network is then trained using the selected hyperparameters and all the training data. The network was tested using the disjointed subset of data selected. These classification accuracies are reported in Tables 9–11.

References

1. Urick, R.J. *Principles of Underwater Sound*, 2nd ed.; McGraw-Hill: Columbus, OH, USA, 1975.
2. Lurton, X. *An Introduction to Underwater Acoustics: Principles and Applications*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2.
3. Williams, D.P. Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 2497–2502.
4. Huo, G.; Wu, Z.; Li, J. Underwater object classification in sidescan sonar images using deep transfer learning and semisynthetic training data. *IEEE Access* **2020**, *8*, 47407–47418. [CrossRef]
5. Howarth, K.; Neilsen, T.B.; Van Komen, D.F.; Knobles, D.P. Seabed Classification Using a Convolutional Neural Network on Explosive Sounds. *IEEE J. Ocean. Eng.* **2021**, *47*, 670–679. [CrossRef]
6. Ge, Q.; Ruan, F.; Qiao, B.; Zhang, Q.; Zuo, X.; Dang, L. Side-scan sonar image classification based on style transfer and pre-trained convolutional neural networks. *Electronics* **2021**, *10*, 1823. [CrossRef]

7. Zhu, P.; Isaacs, J.; Fu, B.; Ferrari, S. Deep learning feature extraction for target recognition and classification in underwater sonar images. In Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, Australia, 12–15 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2724–2731.
8. Einsidler, D.; Dhanak, M.; Beaujean, P.P. A deep learning approach to target recognition in side-scan sonar imagery. In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4.
9. McKay, J.; Gerg, I.; Monga, V.; Raj, R.G. What's mine is yours: Pretrained CNNs for limited training sonar ATR. In Proceedings of the OCEANS 2017-Anchorage, Anchorage, AK, USA, 18–21 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–7.
10. Kubicek, B.; Sen Gupta, A.; Kirsteins, I. Sonar target representation using two-dimensional Gabor wavelet features. *J. Acoust. Soc. Am.* **2020**, *148*, 2061–2072. [CrossRef] [PubMed]
11. Williams, D.P. On the use of tiny convolutional neural networks for human-expert-level classification performance in sonar imagery. *IEEE J. Ocean. Eng.* **2020**, *46*, 236–260. [CrossRef]
12. Wang, X.; Jiao, J.; Yin, J.; Zhao, W.; Han, X.; Sun, B. Underwater sonar image classification using adaptive weights convolutional neural network. *Appl. Acoust.* **2019**, *146*, 145–154. [CrossRef]
13. Wang, H.; Gao, N.; Xiao, Y.; Tang, Y. Image feature extraction based on improved FCN for UUV side-scan sonar. *Mar. Geophys. Res.* **2020**, *41*, 1–17. [CrossRef]
14. Ferguson, E.L.; Ramakrishnan, R.; Williams, S.B.; Jin, C.T. Convolutional neural networks for passive monitoring of a shallow water environment using a single sensor. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2657–2661.
15. Choi, J.; Choo, Y.; Lee, K. Acoustic classification of surface and underwater vessels in the ocean using supervised machine learning. *Sensors* **2019**, *19*, 3492. [CrossRef] [PubMed]
16. Miao, Y.; Zakharov, Y.V.; Sun, H.; Li, J.; Wang, J. Underwater Acoustic Signal Classification Based on Sparse Time–Frequency Representation and Deep Learning. *IEEE J. Ocean. Eng.* **2021**, *46*, 952–962. [CrossRef]
17. Cinelli, L.; Chaves, G.; Lima, M. Vessel classification through convolutional neural networks using passive sonar spectrogram images. In Proceedings of the Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2018), Armação de Buzios, Brazil, 16–19 September 2018; pp. 21–25.
18. Vahidpour, V.; Rastegarnia, A.; Khalili, A. An automated approach to passive sonar classification using binary image features. *J. Mar. Sci. Appl.* **2015**, *14*, 327–333. [CrossRef]
19. Luo, X.; Zhang, M.; Liu, T.; Huang, M.; Xu, X. An Underwater Acoustic Target Recognition Method Based on Spectrograms with Different Resolutions. *J. Mar. Sci. Eng.* **2021**, *9*, 1246. [CrossRef]
20. Domingos, L.C.; Santos, P.E.; Skelton, P.S.; Brinkworth, R.S.; Sammut, K. A survey of underwater acoustic data classification methods using deep learning for shoreline surveillance. *Sensors* **2022**, *22*, 2181. [CrossRef]
21. Bianco, M.J.; Gerstoft, P.; Traer, J.; Ozanich, E.; Roch, M.A.; Gannot, S.; Deledalle, C.A. Machine learning in acoustics: Theory and applications. *J. Acoust. Soc. Am.* **2019**, *146*, 3590–3628. [CrossRef] [PubMed]
22. Das, A.; Rad, P. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv* **2020**, arXiv:2006.11371.
23. Richard, G.; Le Caillec, J.M.; Habonneau, J.; Gueriot, D. A Deep SAS ATR explainability framework assessment. In Proceedings of the OCEANS 2021: San Diego–Porto, San Diego, CA, USA, 20–23 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
24. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
25. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
26. Panwar, H.; Gupta, P.; Siddiqui, M.K.; Morales-Menendez, R.; Bhardwaj, P.; Singh, V. A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. *Chaos Solitons Fractals* **2020**, *140*, 110190. [CrossRef]
27. Zhang, Y.; Hong, D.; McClement, D.; Oladosu, O.; Pridham, G.; Slaney, G. Grad-CAM helps interpret the deep learning models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging. *J. Neurosci. Methods* **2021**, *353*, 109098. [CrossRef]
28. Kubicek, B.; Gupta, A.S.; Kirsteins, I. Feature Engineering and Classification of Elastic Waves from Partial Wave Simulations of Active Sonar Targets. In Proceedings of the OCEANS 2022, Hampton Roads, VA, USA, 17–21 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5.
29. The Mathworks, Inc. *MATLAB Version 9.12.0.1884302 (R2022a)*; The Mathworks, Inc.: Natick, MA, USA, 2022.
30. Williams, K.L.; Marston, P.L. Backscattering from an elastic sphere: Sommerfeld–Watson transformation and experimental confirmation. *J. Acoust. Soc. Am.* **1985**, *78*, 1093–1102. [CrossRef]
31. Kargl, S.G.; Marston, P.L. Ray synthesis of Lamb wave contributions to the total scattering cross section for an elastic spherical shell. *J. Acoust. Soc. Am.* **1990**, *88*, 1103–1113. [CrossRef]
32. Kargl, S.G.; Marston, P.L. Observations and modeling of the backscattering of short tone bursts from a spherical shell: Lamb wave echoes, glory, and axial reverberations. *J. Acoust. Soc. Am.* **1989**, *85*, 1014–1028. [CrossRef]

33. Gaunaurd, G.; Überall, H. RST analysis of monostatic and bistatic acoustic echoes from an elastic sphere. *J. Acoust. Soc. Am.* **1983**, *73*, 1–12. [CrossRef]
34. Gaunaurd, G.; Werby, M. Lamb and creeping waves around submerged spherical shells resonantly excited by sound scattering. *J. Acoust. Soc. Am.* **1987**, *82*, 2021–2033. [CrossRef]
35. ToolBox, E. Solids and Metals—Speed of Sound. Available online: https://www.engineeringtoolbox.com/sound-speed-solids-d_713.html (accessed on 7 September 2022).
36. Abraham, D.A.; Willett, P.K. Active sonar detection in shallow water using the Page test. *IEEE J. Ocean. Eng.* **2002**, *27*, 35–46. [CrossRef]
37. Mallat, S. *A Wavelet Tour of Signal Processing*; Elsevier: Amsterdam, The Netherlands, 1999.
38. Narayan, R. *Encyclopedia of Biomedical Engineering*; Elsevier: Amsterdam, The Netherlands, 2018.
39. Cao, X.; Ren, L.; Sun, C. Research on obstacle detection and avoidance of autonomous underwater vehicle based on forward-looking sonar. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**. [CrossRef]
40. Qin, X.; Luo, X.; Wu, Z.; Shang, J. Optimizing the sediment classification of small side-scan sonar images based on deep learning. *IEEE Access* **2021**, *9*, 29416–29428. [CrossRef]
41. Santurkar, S.; Tsipras, D.; Ilyas, A.; Madry, A. How does batch normalization help optimization? In Proceedings of the 32nd Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
42. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
43. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Chapter 7, pp. 220–226. Available online: http://www.deeplearningbook.org (accessed on 15 January 2023).
44. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 25.
45. Cho, H.; Kim, Y.; Lee, E.; Choi, D.; Lee, Y.; Rhee, W. Basic enhancement strategies when using Bayesian optimization for hyperparameter tuning of deep neural networks. *IEEE Access* **2020**, *8*, 52588–52608. [CrossRef]
46. Walker, S.; Peeples, J.; Dale, J.; Keller, J.; Zare, A. Explainable Systematic Analysis for Synthetic Aperture Sonar Imagery. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 2835–2838.
47. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
48. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
49. Zhang, L.; Zhang, L.; Mou, X.; Zhang, D. FSIM: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **2011**, *20*, 2378–2386. [CrossRef]