Classification of Imaging Artifacts in Synthetic Aperture Sonar With Bayesian Deep Learning

Marko Orescanin, *Senior Member, IEEE*, Derek Olson, *Senior Member, IEEE*, Brian Harrington, Marc Geilhufe, Roy Edgar Hansen, *Senior Member, IEEE*, Dalton Duvio, and Narada Warakagoda

***Abstract*—Synthetic aperture sonar (SAS) provides high-resolution underwater imaging but can suffer from artifacts due to environment or navigation errors. This work explores Bayesian deep learning for classifying common imaging artifacts while quantifying model reliability. We introduce a novel labeled data set with simulated imaging errors through controlled beamforming perturbations. Two Bayesian neural network variants, Monte Carlo dropout and flipout, were trained on this data to detect three artifacts induced by: sound speed errors, yaw attitude error, and additive noise. Results demonstrate these methods accurately classify artifacts in SAS imagery while producing well-calibrated uncertainty estimates. Uncertainty tends to be higher for uniform seafloor textures where artifacts are harder to perceive, and lower for richly textured environments. Analyzing uncertainty reveals regions likely to be misclassified. By discarding 20% of the most uncertain predictions, classification improves from 0.92 F_1-score to 0.98 F_1-score. Overall, the Bayesian approach enables uncertainty-aware perception, boosting model reliability—an essential capability for real-world autonomous underwater systems. This work establishes Bayesian deep learning as a robust technique for uncertainty quantification and artifact detection in SAS.**

***Index Terms*—Active sonar, Bayesian deep learning (BDL), imaging artifacts, machine learning, synthetic aperture sonar (SAS).**

I. Introduction

HIGH resolution sonar systems are a common tool for seabed remote sensing, including seabed fauna density [1], marine archaeology [2], and automatic target detection (ATR) [3], [4]. Synthetic aperture sonar (SAS) systems collect scattering data on the seafloor over a variety of spatial locations, and are coherently combined [5], [6] to form an image with higher resolution in the along-track direction than is possible using the physical aperture (if frequency and bandwidth are held constant). SAS techniques thus require a high degree of accuracy in the measurement of the position of the sonar platform, its attitude, and the sound speed of the ocean medium [5], [7]. If these requirements are not met, then the resulting beamformed image results in artifacts, such as widening of the point spread function (blurring), periodic aperture errors (grating lobes), and increase in the background noise level [decrease in image signal to noise ratio (SNR)] [8].

There are several methods in the literature that address how to handle poor quality images. These can be classified as either autofocus (which seeks to provide a correction to the image to provide better focus [9], [10]), and image quality assessment [11] (which seeks to determine whether or not a particular image is suitable for a down-stream application). Recent work on image quality mostly has relied on finding strong, small point scatterers in sonar images [12], [13], [14], and measuring their width compared to the diffraction limit [15]. These techniques have two primary drawbacks: 1) speckle is always imaged at the diffraction limit of the aperture (thereby biasing the focus estimate towards better performance), and 2) strong scatterers tend to be larger than a resolution cell, and may not be good candidates for measuring the system's degree of focus (biasing the focus estimate towards worse performance). Complex seafloors tend to have regions of large, high amplitude scatterers [14], such as gravel, and rocky outcrops [16], [17].

To address these deficiencies, we propose using an alternative approach of training a classifier to determine whether an image is corrupted by a particular artifact. In this work, a novel perturbed data set is developed whereby images with good focus are corrupted with several types of image artifacts, following the imaging physics exactly. Then, a deep neural network (DNN) is trained to recognize whether an image has been perturbed by a given error, effectively detecting presence or absence of specific imaging artifacts using a single label as a decision. These images were perturbed with only a single type of error as a first step to assessing image quality of real data. A more realistic approach would be to use a multilabel multiclass classifier, since field data may contain many types of image artifacts. The simpler model was adopted here to answer the question of whether a machine model could reliably detect the presence of specific imaging errors.

To make reliable decisions, models should output well-calibrated uncertainty measures along with predictions. While traditional deterministic neural networks do output probabilities for each class, these probabilities often poorly represent true predictive uncertainty [18]. In contrast, Bayesian neural networks

Received 4 January 2025; revised 17 October 2024; accepted 6 December 2024. This work was supported by the NPS ONR under Grant N0001422WX01861. *(Corresponding author: Marko Orescanin.)*

Associate Editor: A. Hunter.

Marko Orescanin, Brian Harrington, and Dalton Duvio are with the Department of Computer Sciences, Naval Postgraduate School, Monterey, CA 93943 USA (e-mail: marko.orescanin@nps.edu).

Derek Olson is with the Department of Oceanography, Naval Postgraduate School, Monterey, CA 93943 USA (e-mail: derek.olson@nps.edu).

Marc Geilhufe, Roy Edgar Hansen, and Narada Warakagoda are with Norwegian Defence Research Establishment (FFI), 2027 Kjeller, Norway (e-mail: marc.geilhufe@ffi.no).

Digital Object Identifier 10.1109/JOE.2025.3538948

(BNNs) are designed to provide better-calibrated probabilities that more accurately reflect model uncertainty [19], [20]. BNNs achieve this by modeling uncertainty in the network parameters themselves, allowing for the quantification of epistemic uncertainty (model uncertainty) in addition to aleatoric uncertainty (data uncertainty). This capability is particularly valuable in tasks like SAS artifact detection, where understanding the model's confidence in its predictions is crucial. BNNs model the uncertainty of each prediction and produce calibrated probabilities (meaning that as high uncertainty samples are discarded, the classification performance with respect to a relevant metric such as F_1-score increases) [21], [22], [23], [24], [25], [26]. This enables the quantification of uncertainty in prediction and ensures the uncertainty is calibrated, which is critical for many remote sensing and real-time autonomous applications. Further, calibrated uncertainty can identify patterns in data and features that may be missed by traditional deep learning algorithms with deterministic weights. We demonstrate that the estimated bulk uncertainty of Bayesian deep learning (BDL) in imaging artifact predictions, quantified through entropy, correlates with the underlying seafloor texture increasing interoperability of model prediction outputs.

The contributions in this work are 1) creation of a novel data set to study imaging artifacts in SAS, 2) use of a DNN to estimate which type of artifact is present in a given image tile, 3) use of BNN to examine the uncertainty of artifact classification, and 4) linking classifier performance to the image scintillation index (SI) as quantitative metric of image texture.

The rest of this article is organized as follows. In Section II, the methodology is presented, including the data set that was created, and a short overview of the BDL technique used. Section III presents the results of this study. Discussion of these results is given in Section IV, and a link between classification performance and image texture is given. Finally, Section V concludes this article.

II. Methodology

A. Data Set

The field data were collected using the HISAS 1032 interferometric SAS system carried by FFI's HUGIN HUS autonomous underwater vehicle (AUV) [27]. Data were collected from different areas, four were in Norwegian waters and one in the Mediterranean Sea off the coast of Italy, with varying terrain and varying unknown seafloor types. The vehicle altitude was between approximately 15 and 30 m. The SAS center frequency is 100 kHz, and the bandwidth is 30 kHz. The SAS images were constructed using the backprojection algorithm, where an estimated render plane based on sidescan bathymetry is used, and micronavigation is applied for navigation correction [5]. The theoretical resolution in the images is approximately 3.5 cm$\times$ 3.5 cm, and the grid resolution is chosen to be 2 cm $\times$ 2 cm. In total, 28 SAS images were selected, all considered to be of good quality, with the exception of one of the test set images, which was obviously corrupted.

To facilitate model development and evaluation, we divided our data set into the following four distinct sets.

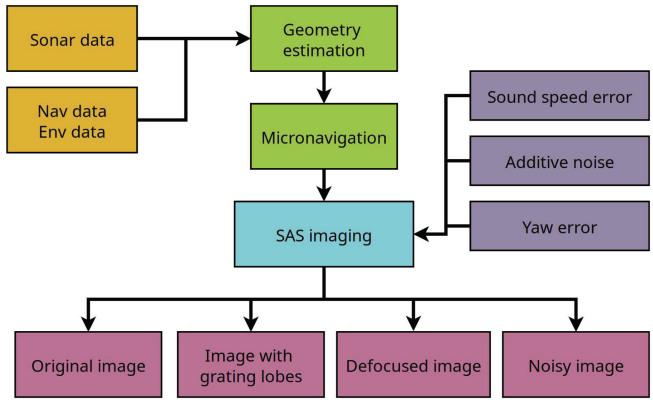


Fig. 1. SAS image degradation scheme. In the diagram, Nav stands for navigation and Env stands for environment.

TABLE I
Data Set Classes

Class number	Description
0	No artifact
1	Yaw error of 0.35°
2	Sound speed error of 40 m/s
3	SNR of 5 dB

1) Training set: Used for model training, comprising the majority of our data.
2) Train-dev set: A subset of training data used to tune hyperparameters and monitor for overfitting.
3) Dev set: Used for model selection and performance evaluation during development.
4) Test set: Comprised of three full images (Test Images I–III) from areas not seen in the other sets, used for final model evaluation and case studies.

In order to produce realistic SAS images of degraded quality, we apply the following procedure (see Fig. 1): We start with the sonar data, the estimated geometry, and the micronavigation solution for the good quality image.

We then either

1) add a sound speed error to be used in the SAS imaging, which will cause defocusing since SAS is nearfield imaging,
2) add a yaw error to the navigation solution to produce uncompensated crab, which will produce periodic errors along the synthetic aperture and thereby grating lobes in the images, and
3) add Gaussian complex noise to the image after imaging and applying range-dependent gain. The noise is modelled to have the same spectral shape as the backscattered signal to simulate the spatial and temporal filtering of the system. The noise is added after imaging in order to approximately obtain range independent SNR reduction.

A thorough discussion of these types of errors can be found in [8]. Together with the original images, this scheme resulted in four different classes listed in Table I. The degradations were serious enough that they could be visually identified as poor quality images by the authors, who primarily work on mine countermeasures and seabed texture. The choice of what degree

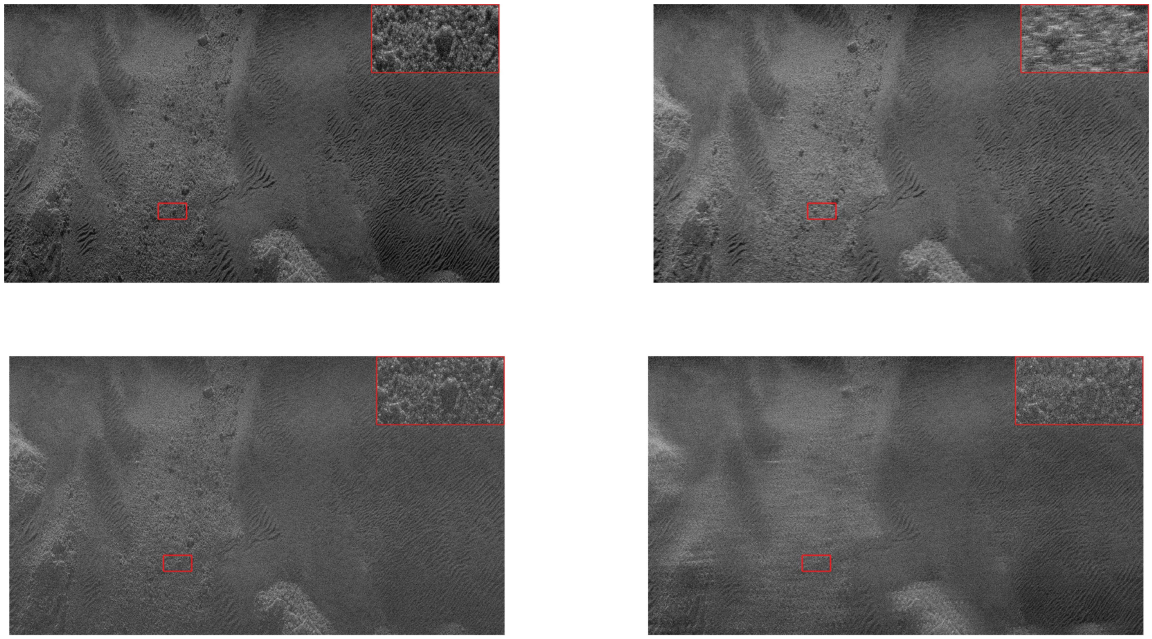


Fig. 2. Example of one of the 28 images from our data set (see Table II, area 3) with and without degradations listed in Table I. The red framed area shows a zoomed in detail of size 10.3 m × 4 m, whereas the total image covers 180 m × 70 m. Top left: original image. Top right: degraded with sound speed error. Bottom left: degraded with noise. Bottom right: degraded with yaw error.

TABLE II
DATA SET OVERVIEW

Area	# images	Image size	Vehicle altitude
1	7	120 m × 120 m	25 m
2	8	120 m × 120 m	15 m
3	10+1	180 m × 70 m	30 m
4	0+1	180 m × 100 m	17 m
5	0+1	180 m × 90 m	25 m
Total	25(+3)		

Three of the total 28 images were set aside as test set. In the number of images column, if there is a plus, the first number corresponds to data that was included in training, and number after the plus corresponds to one of the three images set aside for analysis.

of perturbation to use is not unique and could be explored in a future study.

Fig. 2 shows an example image from the data set without any artifacts as well as each of the three degradations.

In Table II, there is a brief overview of the data. We log scale the acoustic intensity and truncate the dynamic range to 60 dB prior to splitting data into images used for model development and model testing. The goal with such data partition is to ensure that the model does not overfit on the underlying spatial texture distributions. Of the 28 total images, three were removed from the overall data set pool and set aside for test set. Two of those three images were from new areas with seafloor type that is not contained in the other 25 images. The remaining 25 images are split into training, train-dev and dev data sets utilizing the following strategy. Each image is split into regions, where there are subregions of 1500 × 1500 pixels (30 m × 30 m) contributing to dev and train-dev data and the rest of the image is contributing to the train data, see Fig. 3. To further diversify splits we utilize two strategies as illustrated in Fig. 3(a) and (b) that we independently evaluate as data sets throughout the manuscript. This schema is then applied over all 25 images used in the training data set. To summarize

1) break dev and train-dev sub-regions [1500 × 1500 pixels (30 m × 30 m) for each subregion, see Fig. 3] into 300 × 300 nonoverlapping pixel patches (6 m × 6 m).
2) break training region per image into 300 × 300 patches such that they are 50% overlapped to maximize the amount of data available for training.
3) combine all of the dev, train-dev, and training patches into dev, train-dev, and training data sets.

The overlap between patches in the dev data (dev sub-regions) was intentionally avoided to maintain the independence of patches for statistical analysis of the model's performance. Similarly, nonoverlapping patches were used for the train-dev data for the same reasons. However, due to the relatively small number of images available, an overlap of 50% was used on the training set only to artificially create additional images. This procedure results in four times as many images as without using overlap. Overall, such an approach resulted in producing a training data set per strategy of approximately 84 k samples and 5 k train-dev and dev samples each. We emphasize that the three images removed from the overall pool of 28 images are not split into subregions and are preserved as is for use test set to enforce independent analysis. Inference on these images, referred to as image I–III in the rest of the manuscript, is conducted on nonoverlapping patches of 300 × 300 pixels (6 m × 6 m) to produce heat maps of predictions and uncertainty analysis.

B. Variational Inference (VI) in BNNs

BNNs combine the ability of DNN to recognize complex patterns and relationships with the principled parameter estimation in probabilistic models. An advantage of such an approach over the standard deterministic neural networks is the ability to

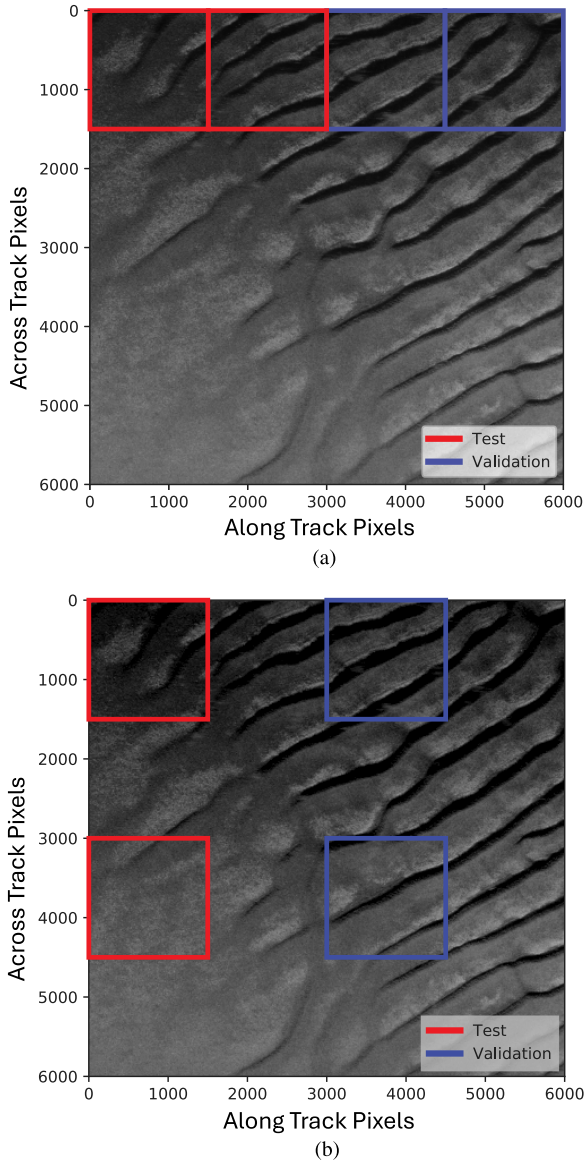


Fig. 3. Illustration of split strategies where image segments extracted for dev and train-dev are outlined in red and blue respectively. Rest of the image is used for training data. In (a) and (b), we depict the specific segments that we utilized over all images regardless of the image size. (a) Split strategy I. (b) Split strategy II.

provide an estimate of uncertainty, and to better self-regularize in training [28], [29]. These aspects are achieved by incorporating a prior probability distribution over the weights of a neural network, $p(\omega)$, with the goal of quantifying a posteriori uncertainty over the network parameters, $p(\omega \mid \mathcal{D})$, given a data set $\mathcal{D}$. This is in contrast to deterministic neural networks where each weight is modeled as a single scalar parameter, ω.

Given a neural network model, $\mathcal{M}$, parameterized by the neural network weights $\omega \in \Omega$, where Ω represents all weights and biases of a neural network architecture, and a supervised learning data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, \ldots, c\}$; N denotes sample size, d is the dimension of the input feature space and c is the number of classes; One can introduce an inference task that needs to be solved

$$p\left(y^*|x^*, \mathcal{D}\right) = \int_{\Omega} p\left(y^*|x^*, \omega\right) p\left(\omega|\mathcal{D}\right) d\omega \tag{1}$$

where x^* is a new input feature vector (e.g., test data), and y^* is the class predicted by the model during inference.

Numerically solving (1) is intractable as one would have to explore an infinite space of posterior forms [20]. As a result, the posterior $p(\omega|\mathcal{D})$ needs to be approximated. VI is one of the methods that can be used to approximate inference and has numerical advantages on large-scale data over sampling methods, such as Markov chain Monte Carlo (MC) [30], [31]. VI involves an optimization approach to approximate $p(\omega|\mathcal{D})$ by fitting an approximation $q_\theta(\omega) \approx p(\omega|\mathcal{D})$ indexed by a variational parameter θ [20]. The intuition behind this approach is that one limits the possible solution space by a family of probability distributions. A member of that family of predetermined distributions $q_\theta(\omega) \in Q$ that is closest to the posterior is found via optimization. Typically, one would measure the distance between the two distributions in terms of Kullback–Leibler (KL) divergence [32], which leads to following formulation of the optimization problem:

$$\theta^\star = \arg\min_{\theta} \Big[\mathrm{KL}[q_\theta(\omega) \;||\; p(\omega|\mathcal{D})]\Big] \tag{2}$$

where

$$\mathrm{KL}[q_\theta(\omega) \;||\; p(\omega|\mathcal{D})] := \int_{\Omega} q_\theta(\omega) \log \frac{q_\theta(\omega)}{p(\omega|\mathcal{D})} d\omega. \tag{3}$$

The minimization problem in (2) can be posed as an optimization problem with the goal of minimizing negative evidence lower bound (ELBO) loss [20]

$$\mathcal{L}_q = \mathrm{KL}[q_\theta(\omega) \;||\; p(\omega)] - \mathbb{E}_q\left[\log p\left(\mathcal{D}|\omega\right)\right] \tag{4}$$

where $\mathbb{E}$ represents the expected value. This loss function encompasses both the data-dependent likelihood loss, and the prior-dependent term that acts as a penalty in optimization. The prior term is also referred to as the complexity cost [33]. For more theoretical details about VI and ELBO we refer the interested reader to Blei et al. [20]. We note that for a deterministic neural network, the negative log-likelihood term in (4) serves a similar purpose to loss functions, such as categorical cross-entropy loss [32] for a multiclass classification problem, or mean-squared error loss for regression with Gaussian output assumption [19]. However, in BNNs, this term appears as an expectation over the approximate posterior and is part of a broader optimization objective (the ELBO) that includes the KL divergence term. The KL divergence acts as a regularizer on the network weights, penalizing deviations of the approximate posterior from the prior. This approach to optimization through ELBO allows BNNs to balance fitting the data with maintaining appropriate uncertainty in the model parameters. Maximizing the ELBO (or minimizing the negative ELBO in practice) encourages both good data fit and a reasonable distribution over model parameters.

The complexity of the optimization of (4) is related to the complexity of the variational family Q as it is easier to optimize

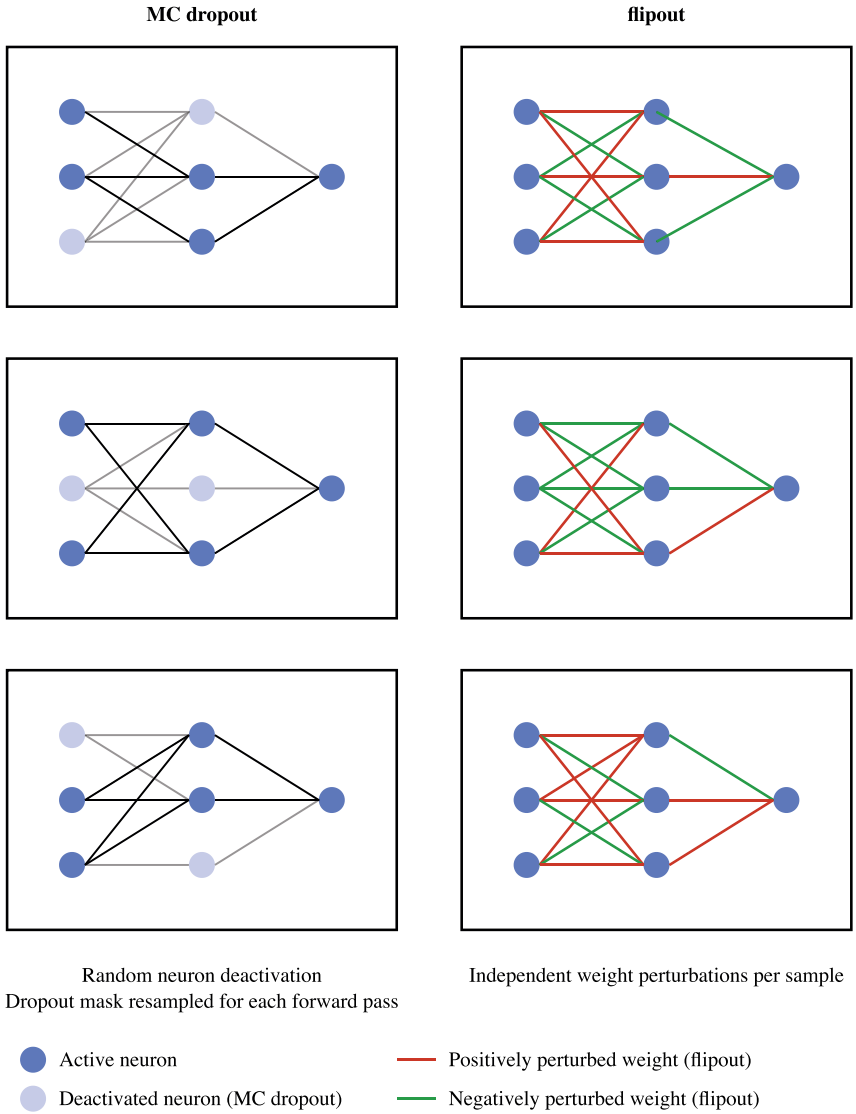


Fig. 4. Comparison of MC dropout and flipout methods. MC dropout randomly deactivates neurons, while flipout perturbs all weights independently for each sample.

over simpler families with fewer parameters, given that the posterior is approximated by $q_\theta(\omega)$ via VI.

C. Bayesian Approximation Methods

In this work, we consider two approaches, a popular Bayesian approximation termed MC dropout, with a Bernoulli prior over the weights [34] and a mean-field approximation via a Gaussian posterior (and prior) with a flipout MC estimator of KL-divergence [35]. Fig. 4 illustrates the key differences between these methods.

1) MC Dropout: MC dropout [34] applies a Bernoulli prior over the weights, randomly deactivating neurons during both training and inference. This approach is widely popular [36] as it requires minimal changes to existing architectures and does not increase the number of model parameters.

During forward passes, MC dropout effectively samples from an approximate posterior over networks by applying dropout masks

$$\mathbf{y} = f(\mathbf{W}(\mathbf{m} \odot \mathbf{x})), \quad \mathbf{m} \sim \text{Bernoulli}(p) \tag{5}$$

where $\mathbf{m}$ is the dropout mask, $\mathbf{W}$ are the weights, $\mathbf{x}$ is the input, f is the activation function, and $\odot$ denotes element-wise multiplication.

In practice, this means that during each forward pass, a random subset of neurons is "turned OFF" according to the dropout probability p. This creates a form of model averaging, as each forward pass effectively uses a different subnetwork. During training, the same dropout mask is applied to all samples in a mini-batch, but different masks are used between mini-batches.

2) Flipout: Flipout [35] uses a mean-field approximation with a Gaussian posterior and prior, employing a MC estimator of the KL-divergence. Unlike MC dropout, flipout creates

pseudoindependent weight perturbations for each example in a mini-batch, decorrelating gradients between samples.

This is achieved by

$$\mathbf{W}_n = \mathbf{W} + \mathbf{\Delta W} \odot (\mathbf{r}_n \mathbf{s}_n^\top) \tag{6}$$

$$\mathbf{y}_n = f(\mathbf{W}_n \mathbf{x}_n) \tag{7}$$

where $\mathbf{\Delta W}$ is a shared perturbation, and $\mathbf{r}_n$ and $\mathbf{s}_n$ are random sign vectors specific to each example n in the mini-batch.

In flipout, each sample in a mini-batch effectively sees a different set of perturbed weights. This is computationally efficient as it allows for parallelization across samples in a mini-batch. The shared perturbation $\mathbf{\Delta W}$ ensures some consistency, while the random sign vectors provide the independence between samples.

While flipout doubles the number of model parameters due to its parameterized distributions, it can achieve better uncertainty calibration than MC dropout [22], [24], [37]. This means the F_1-score increases more rapidly as high-uncertainty predictions are discarded. In addition, flipout's gradient decorrelation improves the conditioning of the Hessian and reduces variance under certain conditions [35].

Flipout has consistently outperformed MC dropout in large-scale remote sensing applications by providing better calibrated uncertainties [21], [22], [24], [37], [38]. For further details on applying these methods to uncertainty quantification in remote sensing tasks, we refer readers to [22], [24], [37], and [38].

D. Uncertainty Quantification With BNNs

For a multiclass classification problem with C classes, the predictive probability based on the last c-dimensional linear output layer of a neural network with parameter vector ω can be represented as $f^\omega = (f_1^\omega \dots f_c^\omega)$. The predictive probability is given by

$$\begin{aligned} p(y^* = c \mid \mathbf{x}^*, \omega) &= p(y^* = c \mid f^\omega(\mathbf{x}^*)) \\ &= \text{softmax}(f^w(\mathbf{x}^*)) \end{aligned} \tag{8}$$

where the softmax function is commonly used to normalize the linear output from a neural network [32] for a multiclass classification task.

For a trained BDL model, prediction uncertainty can be calculated by marginalizing over the approximate posterior distribution $q_\theta(\omega)$ using MC integration [39] with M samples to calculate the mean predictive probability from (1)

$$\begin{aligned} p(\hat{y} = c \mid \boldsymbol{x}^*, \mathcal{D}) &\approx \int p(\hat{y} = c \mid \boldsymbol{x}^*, \omega) q_\theta(\omega) d\omega \\ &\approx \frac{1}{M} \sum_{m=1}^{M} p(\hat{y} = c \mid \boldsymbol{x}^*, \hat{\omega}_m) \\ &\approx \frac{1}{M} \sum_{m=1}^{M} \hat{p}_{c_m} = \bar{p}_c \end{aligned} \tag{9}$$

where $\hat{\omega}_m \sim q_\theta(\omega)$ and c represents the true class (e.g., "Yaw" error or "No Artifact"). A single class is assigned based on (9) and the highest mean predictive probability. Intuitively, in inference we sample from the model weights for the given input multiple times producing an ensemble of predictions, with the final prediction being an ensemble average.

TABLE III
OVERALL NETWORK ARCHITECTURE UTILIZING CUSTOM RESNET BLOCKS

Layer Group	Output Size	Description
Input	300 x 300 x 1	Grayscale image
ResBlock Group 1	300 x 300 x 16	3x ResBlock (16 filters)
ResBlock Group 2	150 x 150 x 32	3x ResBlock (32 filters)
ResBlock Group 3	75 x 75 x 64	3x ResBlock (64 filters)
Output	4	Global Avg Pool, Dense(4)

Each ResBlock can be implemented as deterministic, MC dropout, or flipout variant as detailed in Figure 5

Uncertainty can be quantified via predictive entropy. Normalized predictive entropy measures the average amount of information contained in the predictive distribution and for a multiclass classification problem is given by

$$H_p^*(\hat{y} \mid \boldsymbol{x}^*) = -\sum_{c=1}^{C} \bar{p}_c \frac{\log \bar{p}_c}{\log C} \tag{10}$$

where C represents the number of all possible classes [40].

E. Architecture Choices and Training Methodology

Rather than focusing on optimizing model architectures for the task of detecting imaging artifacts with SAS we adopt a well understood ResNet 20 model to conduct our experiments [41], [42]. To provide a comprehensive view of our model architecture, we present the overall network structure in Table III. This table details the layers and operations used in our ResNet-20 based model. The input to our model is a $300 \times 300 \times 1$ grayscale image patch, which is processed through a series of residual blocks as shown.

For our Bayesian variants, we modify the standard ResNet blocks as illustrated in Fig. 5. This figure shows the architectural differences between the deterministic, MC dropout, and flipout implementations of our residual blocks.

Our model processes each 300×300 pixel image patch independently, effectively operating as a region-based classifier. For a full SAS image, we divide it into nonoverlapping 300×300 pixel regions. The output layer of our model consists of a dense layer with 4 units, corresponding to our four classification categories (no artifact, sound speed, SNR, Yaw). This is followed by a softmax activation function to produce class probabilities. These architectural choices allow our model to effectively capture the complex features present in SAS imagery while providing the flexibility needed for our Bayesian implementations.

We have previously demonstrated in applications that both residual network (ResNet) architectures [41] and custom convolutional neural network (CNN) models can work well in BDL configurations [21], [22], [38] for remote sensing application and on large scale data sets with both MC dropout and flipout approaches.

ResNet architectures for Bayesian experiments were adopted in identical configuration as the deterministic architecture by following the approach in [43]. Moreover, we adopt the identical configuration of Bayesian ResNet blocks as depicted in our previous work by Ortiz et al. [37]. Flipout convolution

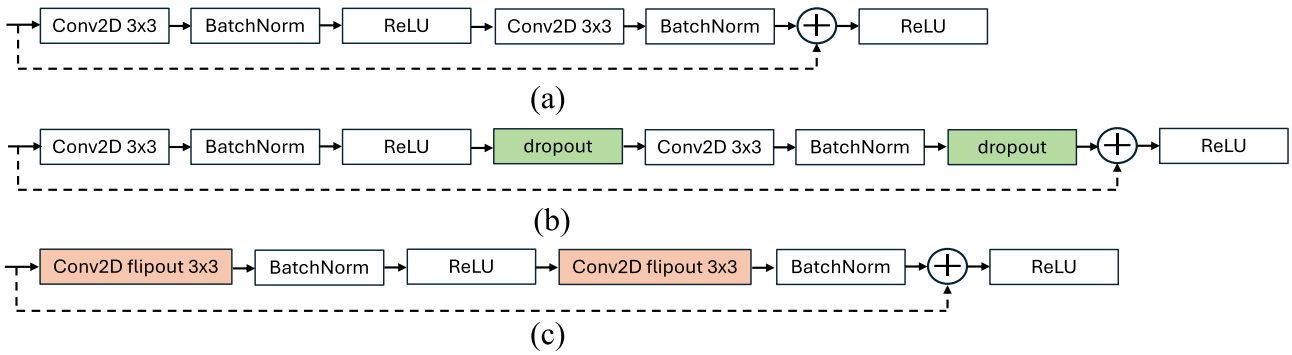


Fig. 5. (a) Illustrates the architecture of a standard deterministic ResNet block. (b) Showcases the modified ResNet block incorporating MC dropout, featuring additional dropout layers following each convolutional operation. These dropout layers remain active during both training and inference stages. Green highlights indicate modifications from the baseline deterministic configuration. (c) Presents the ResNet block adapted for the flipout method, where conventional convolutional layers are substituted with their flipout counterparts. Orange highlights indicate modifications from the baseline deterministic configuration.

layers [35] were implemented utilizing the Tensorflow Probability library [44]. We implemented MC dropout following the approach of Nado et al. [45], inserting dropout layers after each activation layer that follows a convolution within the residuals of the original deterministic model. The exception is the convolution layer immediately preceding a skip connection, where no dropout is applied. Importantly, dropout is retained during both training and inference [19].

All of the developed models used the same training strategy in order to provide a fair comparison of the performance. The model weights were initialized using He initialization [46]. Batch size was set at 128 due to memory limitations. We used the Adam optimizer [47], with starting learning rate of 0.001 and default parameters in TensorFlow.

Learning rate annealing was implemented through monitoring of the validation loss. The rate was reduced by a factor of five if there was no improvement in validation loss within 50 epochs. In order to regularize for overfitting we employed an early stopping strategy [32]. If early stopping did not occur, a model would train for a total of 2000 epochs.

We employed batch normalization in our network architecture, which is a standard practice in deep learning to stabilize training and improve generalization. Batch normalization uses a running mean and variance estimate per pixel, where the ensemble is the batch of data being processed. This type of normalization is useful for standardizing the data for input into deep learning models. While normalization techniques can be beneficial, it's important to note that SAS data has unique characteristics due to imaging geometry, acoustic propagation, and beamforming processes. For example, the variance of the SAS intensity typically increases as a function of range [48], [49], and batch normalization would remove this trend within a patch. This type of normalization could also introduce artifacts if a given batch contained systematic variations in acoustic intensity that are not present in the entire data set (such as discontinuities in seafloor scattering strength). If a relatively homogeneous patch is normalized using a running mean estimated from strongly textured images, the resulting normalized image may have a spatially varying mean and variance. It is standard practice in deep learning to randomize the creation of batches from the data set to avoid this problem, and we follow this practice.

Even though the normalization technique used here has drawbacks, our results demonstrate that the model successfully learned meaningful features and relationships from the SAS data. As discussed below (see Section III-C), the strong correlation between our model's performance and the underlying physical attributes of the seafloor texture, as evidenced by the relationship between uncertainty estimates and the SI, suggests that the network effectively captured relevant SAS-specific patterns and artifacts. The randomized batches have to some extent dealt with the problem of statistically non stationary normalization coefficients. Future work could explore SAS-tailored normalization methods (such as constant false alarm rate normalizers), or texture-specific coefficients that account for the physical realities of acoustic imaging and heterogeneous seafloor environments.

In our previous work [50], we explored various data augmentation techniques, both standard computer vision augmentations as well as pseudocoloring [51], for a SAS image classification task. We selected only along-track flip technique for the data augmentation in our experiments presented here, based on the findings in [50]. This choice was informed by a comprehensive study we conducted on various data augmentation techniques for SAS imagery, including both computer vision and SAS-specific methods. Key findings showed that augmentations preserving SAS imaging physics generally improved model performance, while those violating physical principles often degraded it. Notably, along-track flipping improved accuracy from 86.7% to 88.2%, while across-track flipping reduced it to 81.0%. Flipping the images along the sonar track direction retains physical consistency and increased model accuracy in the prior experiments. This approach aligns with recommendations from other literature in the sonar community [3], [51], [52]. Interestingly, some nonphysical augmentations like zoom ($\pm$ 20%) showed surprising benefits, improving accuracy to 92.0%. However, for this study, we focused on the physically consistent along-track flip to maintain the integrity of the SAS imaging process. For a full analysis of augmentation effects on SAS artifact detection, readers are referred to our detailed study [50].

All of the models were trained using distributed training on four NVIDIA V100 GPUs. Flipout models training time was roughly doubled compared to deterministic models which

is consistent with the findings in [35]. MC dropout models were comparable in training to the deterministic counterparts, with less than 10% difference on averaged epoch run time mid training.

F. Model Selection

Typically for a multiclass classification task [32] when training multiple models one can select among using accuracy (Acc), precision (Prec), recall (Rec), and F_1-score. For the Bayesian models an additional commonly used selection metric is the mean negative log likelihood (MNLL): [53], [54], [55], [56]

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$F_1 = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$$

$$\text{MNLL} = -\left(\frac{1}{N}\sum \log(p(x_c))\right) \tag{11}$$

where TP and TN are true-positives and true-negatives, FP and FN are false-positives and false-negatives, N is the number of samples in the dev data set, and $p(x_c)$ is the probability the model assigns to the true class label.

Accuracy measures the fraction of correct predictions out of the total predictions on a data set. However, for imbalanced data sets, accuracy can be misleading, so precision, recall, and F_1-score better assess model performance. Notably, our data set is class-balanced, so accuracy equals the F_1-score. Precision describes the fraction of predicted positives that are actually correct, while recall is the fraction of samples from each class correctly classified. The F_1-score combines precision and recall as their harmonic mean, with 1 being the best and 0 the worst. For Bayesian model selection, we use MNLL. Since it is desired for models to generalize to new data, MNLL compares the model's predicted distribution to the true distribution on an independent dev set. Lower MNLL indicates the model distribution is closer to the underlying dev data distribution [57]. We used the scikit-learn library [58] in order to calculate model selection metrics.

III. Results

We first examine the performance of the deterministic ResNet architecture on two split strategies, see Fig. 3. To regularize for overfitting and to expand the data set we augmented the data by applying along-track flip, a common data augmentation used on SAS images [3]. Performance between the two split strategies were comparable, see Tables IV and V, with Split Strategy II slightly better than split strategy I. Overall, the average F_1-score for Split Strategy I was 0.884 and 0.897 for split strategy II. Per class performance is comparable between the two strategies as well. Here, the lowest performing skill is the estimation of Yaw artifacts, with split strategy I producing an F_1-score of 0.798 and split strategy II achieving F_1-score of 0.805. Going forward with the analysis we focused only on split strategy I which provides a marginally better baseline for a generalized image training approach given that if the images cannot be split evenly as above for split strategy II, see Fig. 3, the locations of the patches may not be trivial to determine. Selecting the top quarter of each image for dev and train-dev is a simpler and more reproducible strategy while providing spatial separation between the dev and train-dev data.

TABLE IV
Splits Strategy I: Deterministic ResNet20

	precision	recall	F_1-score	support
No Artifact	0.866	0.782	0.822	1250
Sound Speed	0.932	0.911	0.922	1250
SNR	1.000	0.992	0.996	1250
Yaw	0.753	0.850	0.798	1250
accuracy	0.884			
macro avg	0.888	0.884	0.884	5000

TABLE V
Split Strategy II: Deterministic ResNet20

	precision	recall	F_1-score	support
No Artifact	0.844	0.815	0.829	1250
Sound Speed	0.990	0.921	0.954	1250
SNR	1.000	1.000	1.000	1250
Yaw	0.767	0.846	0.805	1250
accuracy	0.896			
macro avg	0.900	0.896	0.897	5000

TABLE VI
Splits Use Test Accuracy Results

Split	Image II accuracy	Image III accuracy
Strategy I	75.9%	91.5%
Strategy II	78.1%	89.7%

TABLE VII
Model Performance

Model	Dev Accuracy	MNLL
Deterministic	90.2%	N/A
MC dropout	87.2%	0.433
flipout	92.7%	0.245

Similar conclusions are drawn from inference on two of the three standalone images where performance statistics are spatially averaged over the whole image (standalone image I has considerable existing errors and ground truth is unknown). As an example, average accuracy per image varies approximately 2% depending on the strategy utilized to develop the splits that the model is trained on, see Table VI.

A. Bayesian Methods

A comparison of the Bayesian methods tested is given in Tables VII and VIII. Flipout performed the best of three methods, having the lowest MNLL and the highest accuracy. This is expected, in part, as a result of how these methods approximate the posterior. As described earlier, MC dropout uses model weights from the deterministic prior, whereas flipout uses an explicitly Gaussian prior.

Flipout is able to reduce the variance between the model and the underlying data distribution. As observed from Table VIII, the MC dropout model presents a large deviation between the

TABLE VIII
PER CLASS MODEL PERFORMANCE

	Deterministic			MC dropout			flipout			
	precision	recall	F_1-score	precision	recall	F_1-score	precision	recall	F_1-score	support
No Artifact	0.816	0.840	0.828	0.700	0.948	0.806	0.871	0.882	0.876	1250
Sound Speed	0.973	0.971	0.972	0.996	0.883	0.936	0.992	0.958	0.975	1250
SNR	1.000	0.994	0.997	1.000	0.994	0.997	1.000	1.000	1.000	1250
Yaw	0.819	0.802	0.810	0.865	0.663	0.751	0.849	0.867	0.858	1250
accuracy		0.902			0.872			0.927		
macro avg	0.902	0.902	0.902	0.891	0.872	0.872	0.928	0.927	0.927	5000

precision and recall values for each class, as much as 0.248. The flipout and deterministic performance is more leveled; the largest deviation in either is 0.034. This highlights the larger variance produced from the MC dropout model. For the above reasons, we chose to focus on the flipout model for detailed uncertainty quantification as we utilize MNLL as a Bayesian model selection criteria.

To assess the statistical significance of our results, we employed a bootstrapping approach using our Bayesian probabilistic models. For each model, we performed 20 iterations, each using 25 randomly selected inferences from a total of 500 available inferences. We calculated the mean accuracy and 95% confidence intervals for these subsets. The confidence intervals were computed using the t-distribution, assuming normality of the sampling distribution of the mean. Our analysis yielded the following results:

1) MC dropout model: mean accuracy of 0.8712 with a 95% confidence interval of (0.8708, 0.8717).
2) Flipout model: mean accuracy of 0.9272 with a 95% confidence interval of (0.9268, 0.9276).

Based on these results, we can conclude that differences in accuracy of 0.1% or greater are statistically significant ($p < 0.05$) for our models. The nonoverlapping confidence intervals between our MC dropout and flipout models indicate a statistically significant difference in their performance.

While we did not calculate confidence intervals for all metrics and tables we expect similar levels of precision across our other performance metrics. Throughout our results, we consider differences of 0.2% or greater as potentially statistically significant. However, we emphasize that statistical significance does not always imply practical significance, especially for very small differences. Readers are encouraged to consider the practical implications of performance differences in the context of specific applications.

B. Uncertainty Calibration

To properly characterize the uncertainty, we begin with an analysis of the relationship between the predictive entropy H_p^* and the F_1-score. This analysis was conducted on our dev set, which serves as our primary evaluation set for model performance and uncertainty calibration. In Fig. 6, we examine how removing (or discarding) the most uncertain data from this dev set improves the F_1-score. By using the dev set for this analysis, we ensure that our uncertainty calibration assessment is performed on data that the model did not see during training or hyperparameter tuning, providing a reliable estimate of the model's generalization performance and uncertainty calibration on new, unseen data. This process is known as verifying uncertainty calibration [22], [23], [59] and is utilized in this work as an additional Bayesian model selection criterion. An ideal curve would asymptotically, and monotonically reach the upper right corner of the plot and would represent a model that has 100% accuracy with very low uncertainty. Given that flipout performance is not ideal we can increase from an F_1-score of 0.92 to an F_1-score of 0.98 by only dropping 20% of its data. MC dropout would need to discard closer to 40% of its data to achieve the same F_1-score.

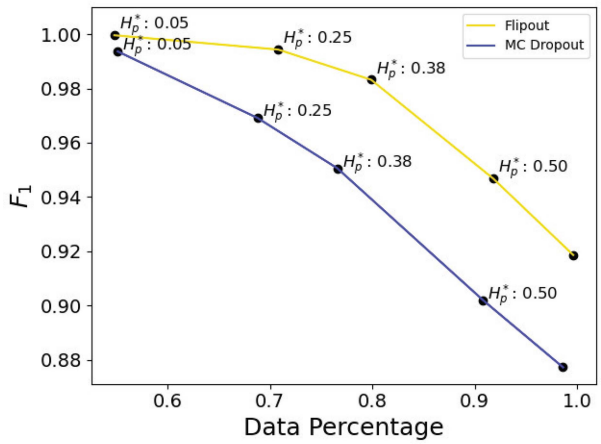


Fig. 6. Plot of calibrated uncertainty for the two different Bayesian models used here. A well calibrated model should have a monotonic dependence between data percentage retained and F_1-score. Flipout has a higher score than dropout for a given data percentage, and therefore is the preferred model in this work.

C. Uncertainty Quantification on a Test Set

To provide a fair assessment of uncertainty, we analyzed the uncertainty on three images that are not part of the model development and Bayesian selection data set, known as the test set here. These images are labeled Images I–III and come from regions 3, 5, and 4, respectively, as described in Table II. Images I and II were collected under good conditions (i.e., similar to the train, train-dev set, and dev set). Image III was collected during environmentally challenging conditions with with multiple severe artifacts. This situation is referred to as a noisy label. Images I and II were rebeamformed such that artifacts were induced as in the first 25 images, and Image III is left as is because of the preexisting artifacts.

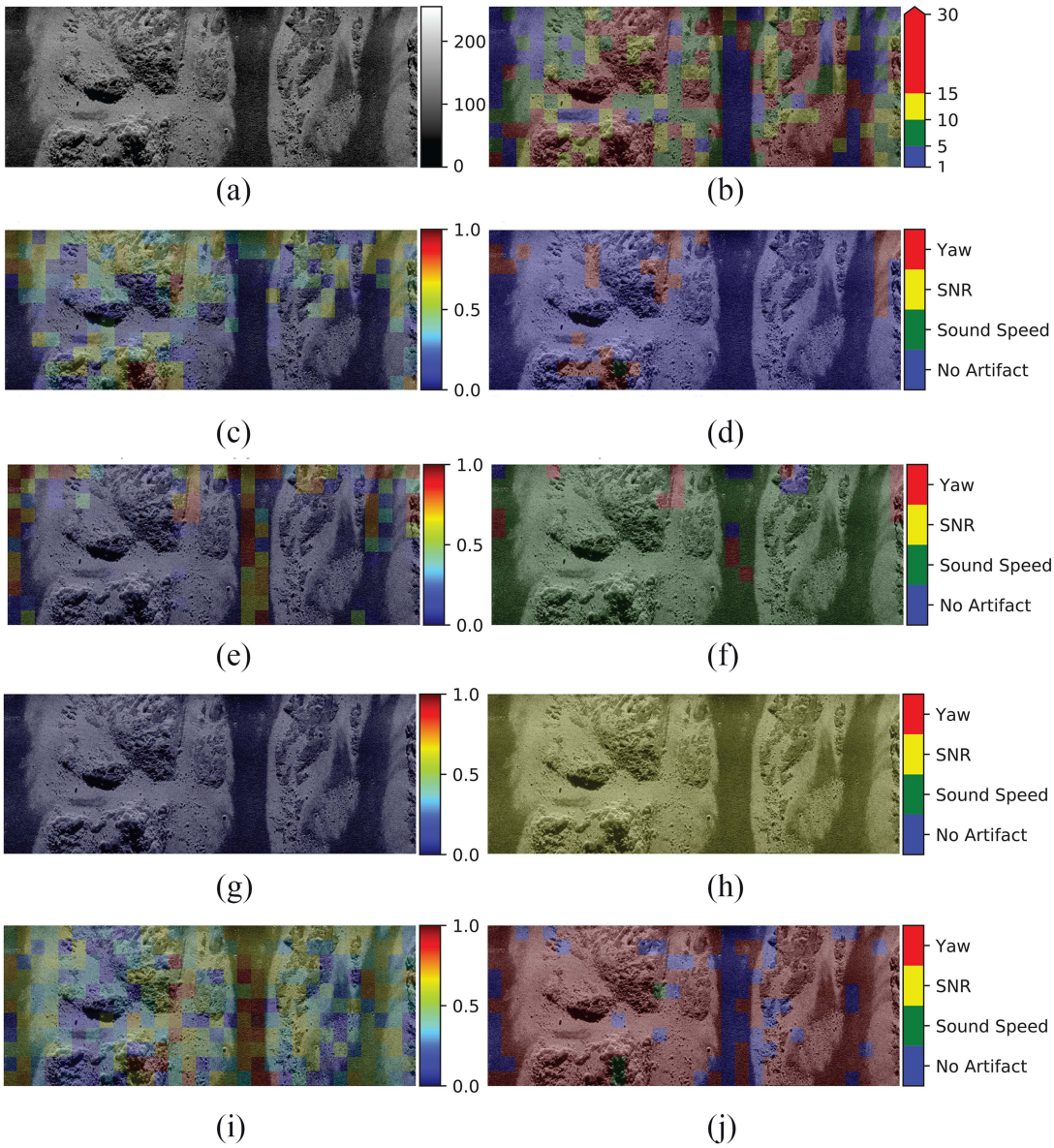


Fig. 7. Test image I: Uncertainty analysis for rocky seafloor area. The image, SI predictions, and entropy for each prediction are shown. (a) Original image. (b) Scintillation index. (c) No Artifact entropy. (d) No artifact predictions. (e) Sound speed entropy. (f) Sound speed predictions. (g) SNR entropy (h) SNR Predictions. (i) Yaw entropy. (j) Yaw predictions. In the images, the horizontal axis is the along track direction in meters and spans 180 m, and vertical is the across track direction spanning 70 m. In (a) the color represents the decibel scaled image, in (b) color represents scintillation index, in (c), (e), (g), and (i) color shows the dimensionless prediction entropy, and in (d), (f), (h), and (j) colors show categorical artifact prediction.

We analyze the test images in terms of the SI that quantifies relative variance in image intensity. SI is defined as follows:

$$\mathrm{SI} = \frac{E[I^2]}{E[I]^2} - 1 \tag{12}$$

where I is an array of the intensity values of a patch of pixels and $E[x]$ is the mean of some quantity x . For this analysis (both for SI and inference), we use nonoverlapping patches of 300 $\times$ 300 pixels, consistent with our approach for the test images as described in Section II-A. These patches correspond to the individual inputs to our neural network and determine the grid resolution in our visualization plots (see Figs. 7–9). This patch size balances capturing sufficient local detail for artifact detection while maintaining computational efficiency. Intuitively, a low SI would correspond to smooth textureless surfaces that consist of mostly independent, exponentially distributed pixel intensities [60, Ch 16]. A high SI can be caused by spatially varying average intensity, such as high-amplitude glints from rock outcrops [16], [17], [61], or from an area with uniform intensity, but which also has a single-point heavy-tailed intensity distribution [48], [49], [62], [63], [64], [65].

At first, we analyze image I, which shows a richly textured rocky bottom separated by a mud channel, see Fig. 7(a). Not surprisingly, SI is low in the homogeneous mud channel and high over strongly textured rock surfaces. Overall, the model performs well over all classes as evident in Fig. 7(d), (f), (h), and (j). Sound speed predictions indicate some misclassification in the mud channel, Fig. 7(f), which is correlated with the high

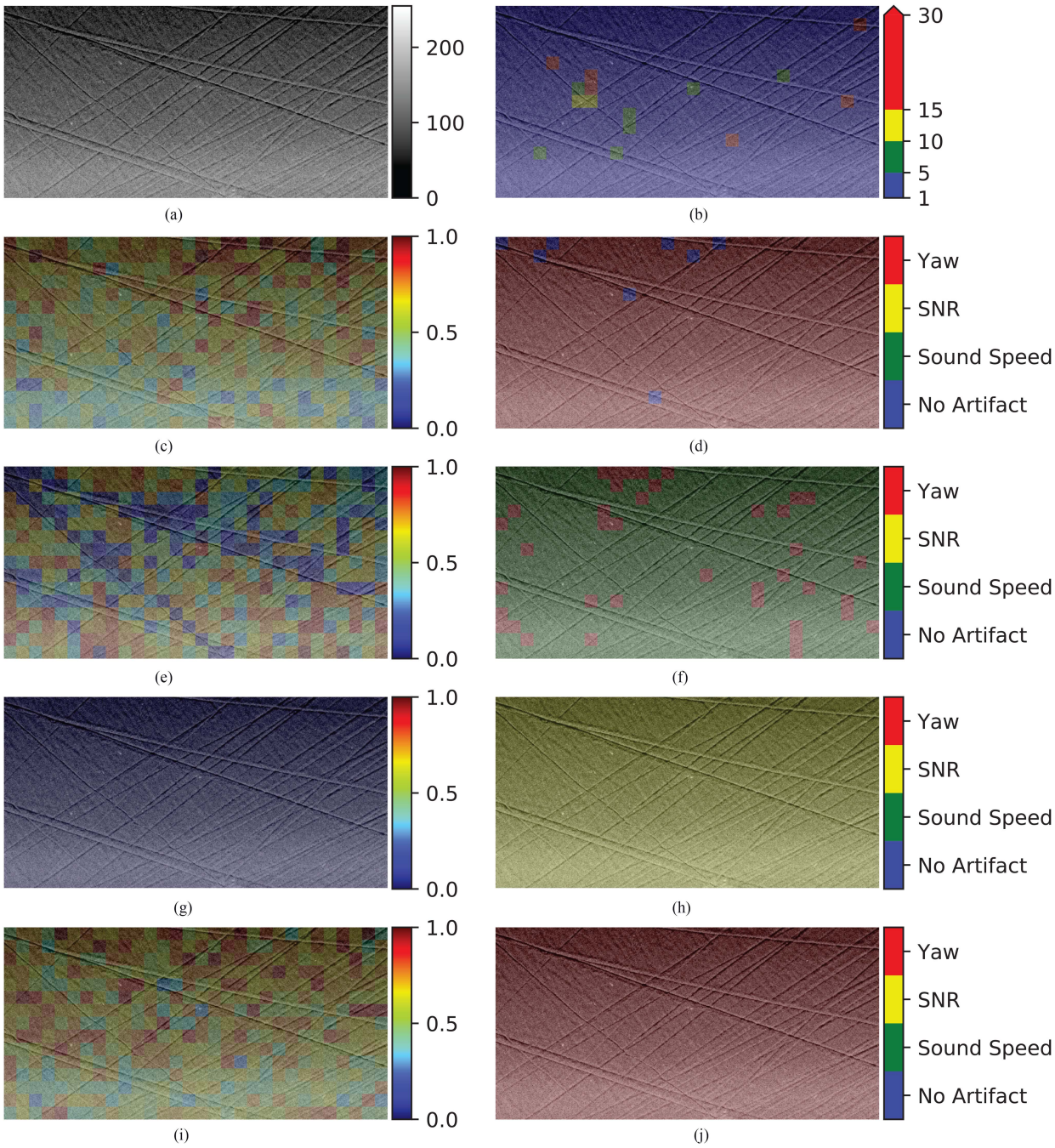


Fig. 8. Image II: Uncertainty analysis for sandy seafloor area. The image, SI predictions, and entropy for each prediction are shown, similar to 7. (a) Original image. (b) Scintillation index. (c) No Artifact entropy. (d) No artifact predictions. (e) Sound speed entropy. (f) Sound speed predictions. (g) SNR entropy (h) SNR Predictions. (i) Yaw entropy. (j) Yaw predictions. In the images, the horizontal axis is the along track direction in meters spanning 180 m, and vertical is the across track direction in meters spanning 90 m. In (a) the color represents the decibel scaled image, in (b) color represents scintillation index, in (c),(e) ,(g), and (i) color shows the dimensionless prediction entropy, and in (d), (f), (h), and (j) colors show categorical artifact prediction.

uncertainties, see Fig. 7(e). Yaw is the most challenging to classify in the mud channel, although the model performs well in the high SI region over rocks. Overall, as can be seen in Fig. 7(c), (e), (g), and (i) the uncertainty trends inversely with the SI for Yaw estimation and can be higher for other classes in the mud channel. In general, we see that patches of high uncertainty lead to an incorrect model prediction.

The next test image is of a sandy bottom with visible trawl lines, image II shown in Fig. 8. Overall, SI is very low, since the seafloor is relatively homogeneous. The trawl lines provide the only source of texture and slightly elevated SI is seen near them. Sound speed, SNR, and Yaw artifacts are overall correctly classified, see Fig. 8(f), (h), and (j), while the No Artifact class is misclassified as Yaw, see Fig. 8(d). Low uncertainty estimates are associated with the SNR artifact prediction, followed by moderate uncertainties on sound speed while high uncertainty estimates are associated with both Yaw and No Artifact classifications. Note that the trawl marks were not seen in the train,

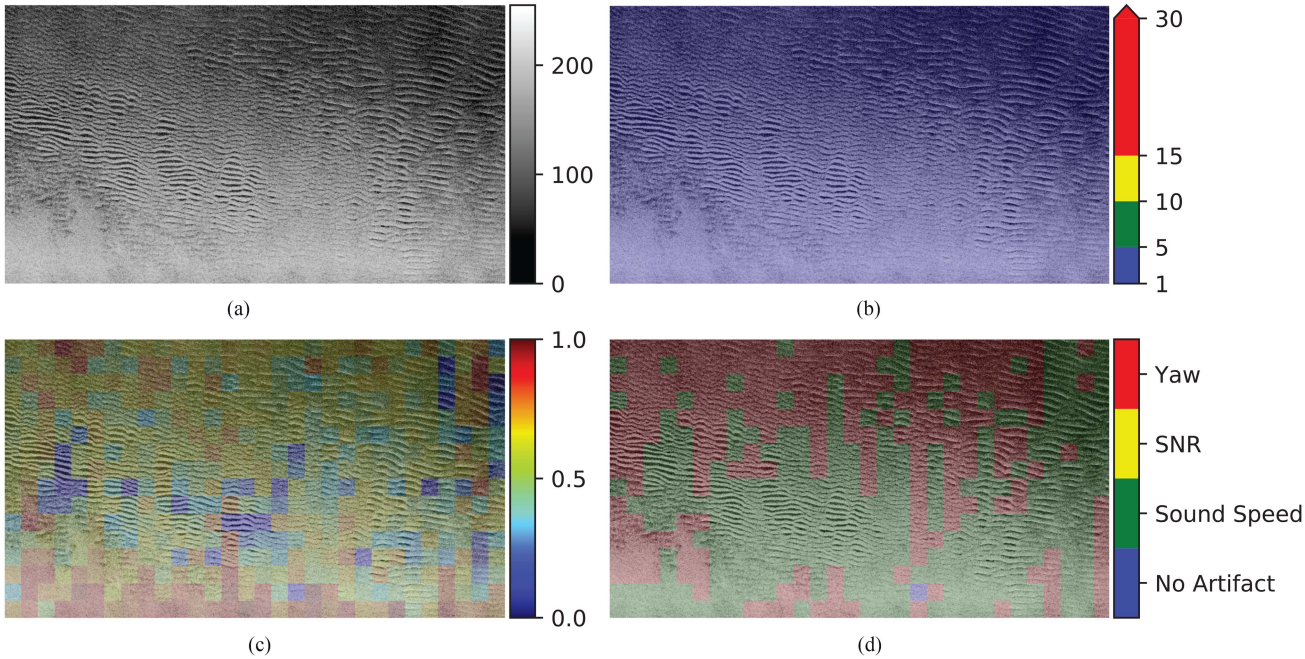


Fig. 9. Image III: Uncertainty analysis for a rippled seafloor with severe existing imaging artifacts (no induced perturbation). Only the overall predictions and entropy are shown for the original image. This image was not perturbed as the other dev images were, due to the existing imaging errors. (a) Original image. (b) Scintillation index. (c) Entropy. (d) Predictions. In the images, the horizontal axis is the along track direction in meters spanning 180 m, and vertical is the across track direction in meters spanning 100 m. In (a) the color represents the decibel scaled image, in (b) color represents scintillation index, in (c), color shows the dimensionless prediction entropy, and in (d) color shows categorical artifact prediction.

train-dev, or dev sets, so this image represents out of distribution test data, and it makes sense that performance is worse compared to image I.

In Image III, see Fig. 9, we examine a rippled, sandy seafloor area collected during challenging surface wave and ocean current conditions (a significant amount of heave, vertical and horizontal crab). This introduced a nonlinear track and uncertainties in the navigational parameters needed for accurate beamforming, meaning that multiple artifacts are likely present in the image. Because of these factors, we do not present accuracy results, and only present the original image, SI, prediction entropy, and the class predictions per patch. Since the Bayesian model classifies most of the image as having yaw and sound speed errors, our single-class classification picks up on the presence of multiple artifacts as fluctuations in the predictions. Our model uncertainty is high throughout the homogeneous region near the bottom of the image where artifacts are hard to perceive. Given the underlying navigational issues, the proper way to address this classification problem would be using a multilabel classifier with potential simultaneous artifacts. Since the single-class classifier has issues with a realistic image with multiple perturbations, we perform a controlled study with multiple induced artifacts in the next section.

D. Multiple Induced Artifacts Case Study

To better understand our model's performance in more complex, realistic scenarios, we conducted a case study investigating the impact of multiple simultaneous artifacts on SAS image quality assessment. We focused on Image I as our base image to introduce multiple artifacts since it contains textures similar to those present in the training set. We introduced one major artifact along with one or two minor artifacts and evaluated our model's performance. For major perturbations, we used sound speed errors of 40 m/s, SNR degradations of 5 dB, or yaw errors of 0.35°. Minor perturbations were set at lower levels: 20 m/s for sound speed, 10 dB for SNR, and 0.2° for yaw. Tables IX–XI present the results for cases where sound speed error, SNR degradation, and yaw error were the primary artifacts, respectively. These two cases represented acceptable versus unacceptable perturbations given the authors experience in target detection and seafloor characterization.

TABLE IX
SOUND SPEED AS A MAJOR ARTIFACT

SoundSpeed [m/s]	SNR [dB]	Yaw [deg]	F_1	Top2Acc	Top2F_1
20			0.65	0.59	0.74
40			0.97	0.98	0.99
40		0.2	0.97	0.98	0.99
40	10		0.00	0.07	0.13
40	10	0.2	0.00	0.1	0.19

Model performance through various combinations of sound speed errors, SNR degradations, and yaw errors are shown. F1-score represents standard single-label performance, while Top2Acc and Top2F1 consider the top two predictions.

We report the standard F_1-score, as well as two additional metrics: Top2Acc (Top-2 Accuracy) and Top2F_1 (Top-2 F_1-score). Top2Acc measures the frequency with which the correct label appears in the model's top two predictions, while Top2F_1 is the F_1-score calculated based on considering both top predictions as correct. These metrics provide insight into the model's performance when dealing with multiple artifacts, as they capture cases where the model identifies the primary artifact as its second most likely prediction. This approach allows

TABLE X
SNR AS A MAJOR ARTIFACT

SoundSpeed [m/s]	SNR [dB]	Yaw [deg]	F_1	Top2Acc	Top2F_1
	5		1.0	1.0	1.0
	5		1.0	1.0	1.0
20	5	0.2	1.0	1.0	1.0

Different levels of SNR degradation are combined with minor sound speed and yaw errors. F1-score represents standard single-label performance, while Top2Acc and Top2F1 consider the top two predictions.

TABLE XI
YAW AS A MAJOR ARTIFACT

SoundSpeed [m/s]	SNR [dB]	Yaw [deg]	F_1	Top2Acc	Top2F_1
		0.2	0.57	0.99	1.0
		0.35	0.92	1.0	1.0
20		0.35	0.57	0.84	0.91
	10	0.35	0.00	0.96	0.98
20	10	0.35	0.00	0.95	0.98

Yaw errors of varying magnitudes are combined with minor sound speed errors and SNR degradations. F1-score represents standard single-label performance, while Top2Acc and Top2F1 consider the top two predictions.

us to evaluate how our single-label classifier performs in more complex, realistic scenarios where multiple artifacts may be present simultaneously.

The results in Tables IX–XI reveal several key insights about our model's performance in multiartifact scenarios. First, the model demonstrates high sensitivity to SNR degradation. When SNR is set to 5 dB, it dominates the prediction regardless of other artifacts present, as evidenced by the drop in the F_1 score to 0 in Tables IX and XI. This result aligns with the basic properties of SAS images, as noise can mask subtle features indicative of other artifacts (such as bright points). Sound speed errors of 40 m/s are well detected even with minor yaw errors present ($F_1 = 0.97$, Table IX), but performance drops significantly for 20 m/s errors ($F_1 = 0.65$). This suggests that the model is more sensitive to larger sound speed errors, which cause more noticeable defocusing in SAS images. Yaw error detection shows inconsistent performance, with good results for isolated 0.2° errors ($F_1 = 0.92$, Table XI) but struggles when combined with other artifacts. This reflects the challenge of detecting yaw errors in SAS images, especially when there are other degradations present. The discrepancies between F_1-scores and Top2 metrics [66], particularly for yaw errors, indicate that the model often identifies the correct artifact, but not always as its top prediction. This suggests that in practical SAS applications, considering the top two predictions might provide more robust artifact detection. Overall, these results underscore the complexity of SAS artifact detection in multierror scenarios and highlight the limitations of our single-label approach, pointing toward the need for a multilabel classification model for real-world SAS applications.

IV. DISCUSSION

From the three test images, a few key results can be gleaned. The SI trends mostly inversely with uncertainty for two classes (Yaw and No Artifact). Patches with low SI can make image quality assessment challenging. This result is in concord with the prior literature on synthetic aperture autofocus, where quantitative metrics for image texture are used as weighting functions for phase estimators [10]. It is possible that a mechanism for learning textural features could be incorporated into our model as an additional source of information that could inform uncertainty.

Inaccurately classified patches often have high uncertainty. This relationship validates the calibrated uncertainty results in Fig. 6, however, results in Fig. 9 raise the question of the impact of texture on uncertainty calibration. We trained models on simulated artifacts, but real-world data can have inherent errors from hardware or environmental issues, as Fig. 9 reveals. BNNs provide the ability to estimate epistemic uncertainty [19] or model uncertainty. This enables identifying potentially inaccurate predictions in challenging areas where artifacts are hard to predict.

The most common confusion by the model is the relationship between Yaw and No Artifact, however, the error correlates well with the uncertainty. The physical effects of a yaw error are difficult to spot by the model. This makes physical sense because uncompensated yaw results in grating lobe copies of strong, isolated targets, and results in a reduced contrast for more homogeneous textures (and can often look like an image with lower SNR). Thus, areas with relatively homogeneous textures are easily confused with areas corrupted by uncompensated yaw. Also, perhaps larger image patches need to be used, since isolated grating lobes may appear outside the tile being analyzed. It may be advantageous to use a traditional signal processing-based approach to estimate yaw, given that its physical effects are relatively well understood [8].

V. CONCLUSION

This work demonstrates the effectiveness of BDL for artifact detection and uncertainty-aware perception using SAS. We generate a labeled data set by emulating common artifacts through precise beamforming perturbations. BNNs provided reliable confidence estimates by modeling uncertainty. This enabled identifying potentially inaccurate predictions in areas where artifacts are hard to perceive.

Our experiments demonstrated that Bayesian techniques can accurately classify emulated artifacts while producing well-calibrated uncertainty estimates. Seafloor texture strongly affected model performance, indicating texture analysis could further improve reliability. By filtering out doubtful predictions, uncertainty awareness enhanced classification accuracy.

Reliable perception is critical for AUV. Bayesian learning offers a pathway to safeguard missions by quantifying model limitations. This work establishes Bayesian networks as a promising technique for explaining model behavior. The uncertainty estimates enable monitoring model reliability even when external factors like weather degrade performance.

The results of this work only apply to a single sensor collected in several different geographic regions. To apply our methods to a new sensor, the ideal case would be to create an a similar data set using the imaging perturbations. However, it may be possible to use transfer learning, beginning with the weights of this model and training the model on data from a different sensor, similar to the work performed by Williams [67] in the context of ATR. A

study to determine the effectiveness of this approach is fruitful area of future research.

One drawback of this study was that a single label classifier was used. In real field data several perturbations are present and future work could use a multilabel classifier. The degree of image perturbations (e.g., a sound speed error of 40 m/s) was chosen using intuition gained by the authors' collective work on target detection and seafloor texture modeling. However, the thresholds for classifying imaging errors by a model designed for this purpose are not yet known, and should be ascertained by future work. In this case, a data set with a wide distribution of imaging errors would have to be created.

In conclusion, this research introduces a principled framework for uncertainty-aware deep learning in SAS. The core technical contributions establish strong baseline capabilities using simulated artifacts. Future work can build upon these results to refine real-world reliability. Overall, BDL provides the transparency and robustness needed for safe autonomy in complex underwater environments.

ACKNOWLEDGMENT

The authors would like to thank the HUGIN AUV operators and researchers at the Norwegian Defence Research Establishment (FFI, Kjeller, Norway) for gathering the data. The authors would also like to thank NATO Centre for Maritime Research and Experimentation (CMRE, La Spezia, Italy) for hosting and organizing trials.

REFERENCES

[1] C. Heinrich, P. Feldens, and K. Schwarzer, "Highly dynamic biological seabed alterations revealed by side scan sonar tracking of Lanice conchilega beds offshore the island of SYLT (German bight)," *Geo- Mar. Lett.*, vol. 37, no. 3, pp. 289–303, 2017.

[2] H. Singh, J. Adams, D. Mindell, and B. Foley, "Imaging underwater for archaeology," *J. Field Archaeol.*, vol. 27, no. 3, 2000, Art. no. 319.

[3] D. P. Williams, "Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks," in *Proc. 23 rd Int. Conf. Pattern Recognit.*, 2016, pp. 2497–2502.

[4] A. Galusha, J. Dale, J. Keller, and A. Zare, "Deep convolutional neural network target classification for underwater synthetic aperture sonar imagery," in *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV*, California, USA: SPIE, 2019, pp. 18–28.

[5] R. E. Hansen, H. J. Callow, T. O. Sæbø, and S. A. V. Synnes, "Challenges in seafloor imaging and mapping with synthetic aperture sonar," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, p. 3677–3687, Oct. 2011.

[6] P. Vouras et al., "An overview of advances in signal processing techniques for classical and quantum wideband synthetic apertures," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 2, pp. 317–369, Mar. 2023.

[7] H. J. Callow, "Signal processing for synthetic aperture sonar image enhancement," Ph.D. dissertation, Univ. Canterbury, Christchurch, New Zealand, Apr. 2003.

[8] D. A. Cook and D. C. Brown, "Analysis of phase error effects on stripmap SAS," *IEEE J. Ocean. Eng.*, vol. 34, no. 3, pp. 250–261, Jul. 2009.

[9] W. G. Carrara, R. S. Goodman, and M. Majewski Ronald, *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms*. Norwood, MA, USA: Artech House, 1995.

[10] I. D. Gerg and V. Monga, "Real-time, deep synthetic aperture sonar (SAS) autofocus," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2021, pp. 8684–8687.

[11] R. E. Hansen and T. O. Saebo, "Towards automated performance assessment in synthetic aperture sonar," in *Proc. MTS/IEEE OCEANS - Bergen*, Jun. 2013, pp. 1–9.

[12] D. J. Pate, D. A. Cook, and B. N. ODonnell, "Estimation of synthetic aperture resolution by measuring point scatterer responses," *IEEE J. Ocean. Eng.*, vol. 47, no. 2, pp. 457–471, Apr. 2022.

[13] M. Geilhufe, R. E. Hansen, Ø. Midtgaard, and S. A. V. Synnes, "Through-the-sensor sharpness estimation for synthetic aperture sonar images," in *Proc. Oceans 2019 MTS/IEEE*, Seattle, WA, USA, Oct. 2019, pp. 1–6.

[14] M. Geilhufe, D. Olson, R. E. Hansen, and S. A. V. Synnes, "A wavelet shrinkage approach to detect candidate point scatterers in synthetic aperture sonar images for resolution estimation," in *Proc. 5th Int. Conf. Synthetic Aperture Sonar Radar, Ser. Inst. Acoust.*. Lerici, Italy, Sep. 2023.

[15] M. Born and E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference, and Diffraction of Light*, 7th ed. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[16] D. R. Olson, A. P. Lyons, and T. O. Sæbø, "Measurements of high-frequency acoustic scattering from glacially eroded rock outcrops," *J. Acoustical Soc. America*, vol. 139, no. 4, pp. 1833–1847, 2016.

[17] D. R. Olson, A. P. Lyons, D. A. Abraham, and T. O. Sæbø, "Scattering statistics of rock outcrops: Model-data comparisons and Bayesian inference using mixture distributions," *J. Acoustical Soc. Amer.*, vol. 145, no. 2, pp. 761–774, 2019.

[18] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn. Ser. Mach. Learn. Res.*, Aug. 2017, pp. 1321–1330. [Online]. Available: https://proceedings.mlr.press/v70/guo17a.html

[19] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 20, pp. 1–11.

[20] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.

[21] M. Orescanin, V. Petković, S. W. Powell, B. R. Marsh, and S. C. Heslin, "Bayesian deep learning for passive microwave precipitation type detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 4500705.

[22] P. Ortiz, M. Orescanin, V. Petkovic, S. W. Powell, and B. Marsh, "Decomposing satellite-based classification uncertainties in large earth science datasets," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4106211.

[23] B. Beckler et al., "Multilabel classification of heterogeneous underwater soundscapes with Bayesian deep learning," *IEEE J. Ocean. Eng.*, vol. 47, no. 4, pp. 1143–1154, Oct. 2022.

[24] J. Fischer, M. Orescanin, P. Leary, and K. B. Smith, "Active Bayesian deep learning with vector sensor for passive sonar sensing of the ocean," *IEEE J. Ocean. Eng.*, vol. 48, no. 3, pp. 837–852, Jul. 2023.

[25] J. Fischer, M. Orescanin, and E. Eckstrand, "VI-PANN: Harnessing transfer learning and uncertainty-aware variational inference for improved generalization in audio pattern recognition," *IEEE Access*, vol. 12, pp. 33347–33360, 2024.

[26] L. Rombado, M. Orescanin, and M. Orescanin, "Uncertainty-aware aerial coastal imagery pattern recognition through transfer learning with imagenet-1 K variational embeddings," *IEEE Access*, vol. 12, pp. 130866–130883, 2024.

[27] T. O. Sæbø, B. Langli, H. J. Callow, E. O. Hammerstad, and R. E. Hansen, "Bathymetric capabilities of the HISAS interferometric synthetic aperture sonar," in *Proc. Oceans MTS/IEEE*, Vancouver, Canada, Oct. 2007, pp. 1–10.

[28] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. R. Wallach, S. Fergus Vishwanathan, and R. Garnett, Eds., vol. 30. Newry, U.K.: Curran Associates, Inc., 2017.

[29] H. Wang and D.-Y. Yeung, "A survey on bayesian deep learning," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–37, Oct. 2020.

[30] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

[31] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, no. 4, pp. 1303–1347, 2013. [Online]. Available: http://jmlr.org/papers/v14/hoffman13a.html

[32] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, vol. 1, no. 2.

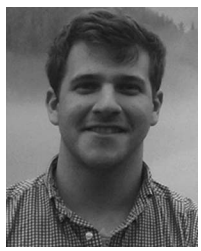
[33] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. 32nd Int. Conf. Mach. Learn., Ser. Mach. Learn. Res.*, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, Jul. 2015, pp. 1613–1622. [Online]. Available: http://proceedings.mlr.press/v37/blundell15.html

[34] Y. Gal, "Uncertainty in deep learning," Dept Eng., Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 2016.

[35] Y. Wen, P. Vicol, J. Ba, D. Tran, and R. B. Grosse, "Flipou: Efficient pseudo-independent weight perturbations on mini-batches," in *Proc. 6th Int. Conf. Learn. Representations*, 2018, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=rJNpifWAb
[36] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6440–6461, Nov. 2018.
[37] P. Ortiz, E. Casas, M. Orescanin, S. W. Powell, V. Petkovic, and M. Hall, "Uncertainty calibration of passive microwave brightness temperatures predicted by Bayesian deep learning models," *Artif. Intell. Earth Syst.*, vol. 2, no. 4, 2023, Art. no. e220056. [Online]. Available: https://journals.ametsoc.org/view/journals/aies/2/4/AIES-D-22-0056.1.xml
[38] B. Beckler et al., "Multi-label classification of heterogeneous underwater soundscapes with Bayesian deep learning," vol. 47, no. 4, pp. 1143–1154, Oct. 2022.
[39] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
[40] L. A. F. Park and S. Simoff, "Using entropy as a measure of acceptance for multi-label classification," in *Advances in Intelligent Data Analysis XIV*, E. Fromont, T. De Bie, and M. van Leeuwen, Eds. Cham, Switzerland: Springer, 2015, pp. 217–228.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.
[43] D. Tran, M. Dusenberry, M. van der Wilk, and D. Hafner, "Bayesian layers: A module for neural network uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14660–14672.
[44] J. V. Dillon et al., "Tensorflow distributions," 2017, *arXiv:1711.10604*.
[45] Z. Nado et al., ""Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning," 2021, *arXiv:2106.04015*.
[46] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Comput. Vis.–ECCV, Ser. Lecture Notes Comput. Sci.*, B. J. LeibeMatas, N. Sebe, and M. Welling, Eds. Cham: Springer, 2016, pp. 630–645.
[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Jan. 2017, *arXiv:1412.6980*.
[48] A. P. Lyons, D. R. Olson, and R. E. Hansen, "Modeling the effect of random roughness on synthetic aperture sonar image statistics," *J. Acoustical Soc. Amer.*, vol. 152, no. 3, pp. 1363–1374, Sep. 2022.
[49] D. R. Olson and A. P. Lyons, "Resolution dependence of rough surface scattering using a power law roughness spectrum," *J. Acoustical Soc. Amer.*, vol. 149, no. 1, pp. 28–48, Jan. 2021.
[50] M. Orescanin, B. Harrington, D. Olson, M. Geilhufe, R. E. Hansen, and N. Warakagoda, "A study on the effect of commonly used data augmentation techniques on sonar image artifact detection using deep neural networks," in *Proc. IGARSS IEEE Int. Geosci. Remote Sens. Symp.*, 2023, pp. 360–363.
[51] N. D. Warakagoda and Ø. Midtgaard, "Transfer-learning with deep neural networks for mine recognition in sonar images," in *Proc. Int. Conf. Underwater Acoust.*, 2020, pp. 115–122.
[52] T. S. Brandes, B. Ballard, S. Ramakrishnan, E. Lockhart, B. Marchand, and P. Rabenold, "Environmentally adaptive automated recognition of underwater mines with synthetic aperture sonar imagery," *J. Acoustical Soc. Amer.*, vol. 150, no. 2, pp. 851–863, 2021.
[53] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
[54] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.
[55] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Mach. Learn.*, vol. 45, no. 2, pp. 171–186, 2001.
[56] O. Durr, B. Sick, and E. Murina, *Probabilistic Deep Learning With Python, Keras and TensorFlow Probability*. Shelter Island, NY, USA: Manning Publications Co., 2020.
[57] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*. Cambridge, MA, USA: MIT press, 2022.
[58] L. Buitinck et al., "API design for machine learning software: Experiences from the scikit-learn project," in *Proc. ECML PKDD Workshop: Lang. Data Mining Mach. Learn.*, 2013, pp. 108–122.
[59] N. Band et al., "Benchmarking bayesian deep learning on diabetic retinopathy detection tasks," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, 2021, pp. 1–40. [Online]. Available: https://openreview.net/forum?id=jyd4Lyjr2iB
[60] D. R. Jackson and M. D. Richardson, *High-Frequency Seafloor Acoustics*. New York, NY, USA: Springer, 2007.
[61] R. C. Gauss, J. M. Fialkowski, D. C. Calvo, R. Menis, D. R. Olson, and A. P. Lyons, "Moment-based method to statistically categorize rock outcrops based on their topographical features," in *Proc. OCEANS - MTS/IEEE Washington*, Oct. 2015, pp. 1–5.
[62] A. P. Lyons and D. A. Abraham, "Statistical characterization of high-frequency shallow-water seafloor backscatter," *J. Acoustical Soc. Amer.*, vol. 106, no. 3, pp. 1307–1315, Sep. 1999.
[63] D. Abraham and A. Lyons, "Novel physical interpretations of K-distributed reverberation," *IEEE J. Ocean. Eng.*, vol. 27, no. 4, pp. 800–813, Oct. 2002.
[64] D. A. Abraham and A. P. Lyons, "Reverberation envelope statistics and their dependence on sonar bandwidth and scattering patch size," *IEEE J. Ocean. Eng.*, vol. 29, pp. 126–137, Jan. 2004.
[65] A. P. Lyons, D. R. Olson, and R. E. Hansen, "Quantifying the effect of random seafloor roughness on high-frequency synthetic aperture sonar image statistics," in *Proc. Inst. Acoust. Conf.: Acoustic Environ. Variability, Fluctuations Coherence*, Cambridge, U.K., 2016, pp. 151–158.
[66] M. M. Lapin Hein and B. Schiele, "Top-k multiclass svm," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Newry, NI, U.K.: Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/0336dcbab05b9d5ad24f4333c7658a0e-Paper.pdf
[67] D. P. Williams, "Deep transfer learning across targets and sensors with synthetic aperture sonar data," in *Proc. Synthetic Aperture Sonar Synthetic Aperture Radar*, Lerici, Italy, Sep. 2023.

Marko Orescanin (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois Urbana-Champaign, Champaign, IL, USA.

He is an Assistant Professor with the Computer Science Department, Naval Postgraduate School, Monterey, CA, USA. Previously, he worked at Bose Corporation in MA, where he led research and advanced development of signal processing and machine learning algorithms for audio and speech enhancement in consumer electronics. As Senior Manager of the AI and Data group, he helped drive innovation in consumer electronics. His research interests include signal processing, machine learning, artificial intelligence, deep learning, acoustics, passive and active sonar, unmanned vehicles, and remote sensing.



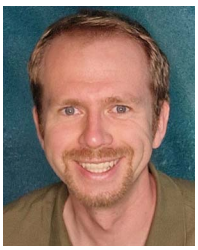
Derek Olson (Senior Member, IEEE) received the AB degree (*cum laude*) in physics from Vassar College, Poughkeepsie, NY, USA, in 2009, and the Ph.D. degree in acoustics from The Pennsylvania State University (PSU), University Park, PA, USA, in 2014.

He is currently an Associate Professor with the Oceanography Department, Naval Postgraduate School (NPS), Monterey, CA, USA. He was previously an Assistant Professor with NPS, and a Research Assistant Professor with the Applied Physics Laboratory, PSU. His research interests include scattering of acoustic waves from natural surfaces, including modeling the physical interactions, as well as measurements of reflections and scattering using synthetic aperture sonar.

Dr. Olson is a Member of the Acoustical Society of America, Phi Beta Kappa, and Sigma Xi. He was the recipient of the Office of Naval Research Young Investigator Award in 2021.

Brian E. Harrington was born and raised in Jacksonville, Florida. He received the bachelor of science in aerospace engineering degree from the United States Naval Academy, Naval Academy, MD, USA, in 2015 with a specialty in astronautical engineering, and the master of science in computer science from Naval Postgraduate School, Monterey, CA, USA, in 2023 and a certificate in Space System Fundamentals.

His capstone project was a future capability assessment of electrical power production, distribution, and requirements for a forward operating base, focused on the viability of space-based solar power. He was a submariner between 2015 and 2020, performing key leadership and technical roles for two overhauls. His thesis, "Image Quality Assessment of Active Sonar Images Through Bayesian Deep Learning," applied probabilistic methodologies to a novel task of detecting imaging artifacts in Synthetic Aperture Sonar images of the ocean floor. He is currently stationed at Supervisor of Shipbuilding – Groton serving as Lead Ship Coordinator.



Marc Geilhufe received the german diploma degree in business mathematics from the Chemnitz University of Technology, Chemnitz, Germany, in 2008, and the Ph.D. degree in statistics from the University of Tromsø, Tromsø, Norway, in 2013.

Since 2013, he has been with the Norwegian Defence Research Establishment (FFI), Kjeller, Norway, working in the field of synthetic aperture sonar. He is currently a Senior Scientist with FFI. His research interests include synthetic aperture sonar, spatial statistics, and image analysis.

Roy Edgar Hansen (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in physics from the University of Tromsø, Tromsø, Norway, in 1992 and 1999, respectively.

From 1992 to 2000, he was with the Norwegian research company TRIAD, Kjeller, Norway, working on multistatic sonar, multistatic radar, SAR, and underwater communications. Since 2000, he has been with the Norwegian Defence Research Establishment (FFI), Kjeller, Norway, working in the field of synthetic aperture sonar. He is currently a Principal Scientist with FFI. He is also an Adjunct Professor in acoustic imaging with the Department of Informatics, University of Oslo, Oslo, Norway. His research interests include synthetic aperture sonar and radar, ultrasound imaging, sonar signal processing, and array signal processing.

Dalton Duvio received the B.A. degree in psychology from Stanford University, in 2016, and the M.S. degree in computer science from Naval Postgraduate School (NPS), in 2023.

He is currently a Research Faculty Associate in the Computer Science Department at NPS. Previously, he conducted neuroscience research at Stanford School of Medicine investigating the efficacy of transcranial magnetic stimulation as an intervention for treatment-resistant depression with precise targeting based on a patient's unique connectome, and the neuroendocrinological aspects of major depression and anxiety. His research interests include machine learning, artificial intelligence, and autonomous agents.

Narada Warakagoda received the M.Sc. and Ph.D. degrees in signal and speech processing from the Norwegian University of Science and Technology, Trondheim, Norway in 1994 and 2001, respectively.

He is a Principal Scientist with Norwegian Defence Research Establishment (FFI). After completion of Ph.D. degree, he started working as a Research Scientist with Telenor Research and Development, Norway. At Telenor, he worked extensively on Speech Recognition algorithms, systems and related areas. Later, his work was focused on software development and service innovations involving areas, such as machine-to-machine communication and computer security. In the period from 2010 to 2015, Warakagoda was with SINTEF ICT, where he contributed to research and development in the area of the Internet of Things. Since 2016, he has been working as a research scientist with FFI in the field of underwater robotics, with an emphasis on image recognition and object detection. His interests include artificial intelligence, machine learning, and software development.