Learned Frequency Domain Masks for Training-Size-Robust Sonar Automatic Target Recognition

Isaac Gerg
Applied Research Laboratory
Pennsylvania State University
University Park, PA 16802
phone: (814) 867-3598 email: idg101@arl.psu.edu

Award Number: N00014-19-1-2563

SUMMARY OF ACCOMPLISHMENTS AND GOALS

Deep learning has enabled significant improvements in semantic image segmentation, especially in underwater imaging domains like side-scan-sonar (SSS). In this work, we apply deep learning to synthetic aperture sonar (SAS) imagery which has an advantage over traditional SSS in that SAS produces coherent high- and constant-resolution imagery. Despite the successes of deep learning, one drawback is the need for abundant labeled training data to enable success. Such abundant labeled data is not always available as in the case of SAS where collections are expensive and obtaining quality ground truth labels may require diver intervention. To overcome these challenges, we propose a domain-specific deep learning network architecture utilizing a unique property to complex-valued SAS imagery: the ability to resolve angle-of-arrival (AoA) of acoustic returns through k-space processing. By sweeping through consecutive incrementally advanced AoA bandpass filters (a process sometimes referred to as multi-look processing), this technique generates a sequence of images emphasizing angle-dependent seafloor scattering and motion from biologics along the seafloor or in the water column. Our proposal, which we call multi-look sequence processing network (MLSP-Net), is a domain-enriched deep neural network architecture that models the multi-look image sequence using a recurrent neural network (RNN) to extract robust features suitable for semantic segmentation of the seafloor without the need for abundant training data. Unlike previous segmentation works in SAS, our model ingests a complex-valued SAS image and affords the ability to learn the AoA filters in k-space as part of the training procedure. We show results on a challenging real-world SAS database, and despite the lack of abundant training data, our proposed method shows superior results over state-of-the-art techniques.

OBJECTIVES

Obtaining labeled training data has been a consistent hurdle for SAS [1], especially for the problem of SAS image segmentation. Many SAS image segmentation methods operate in an unsupervised fashion in order to make research progress despite this hurdle. However, current methods easily confuse classes resulting in mediocre performance on real-world datasets or presenting good efficacy on only one or two classes.

Obtaining accurately labeled imagery for SAS image segmentation necessitates the need for divers and oceanographers to accurately characterize the imagery post-survey. This task is quite burdensome as

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
08-31-2022	Final	01 Aug 2019 to 31 Aug 2022

4. TITLE AND SUBTITLE
Learned Frequency Domain Masks for Training-Size-Robust Sonar Automatic Target Recognition

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-09-1-2563

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Isaac Gerg

5d. PROJECT NUMBER
27512

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Pennsylvania State University Applied Research Labotatory
Office of Sponsored Programs
110 Technology Center Building
University Park, PA 16802-7000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
875 North Randolph Street
Arlington, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)
321

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT
"Approved for Public Release; Distribution is Unlimited."

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Deep learning has enabled significant improvements in semantic image segmentation, especially in underwater imaging domains like side-scan-sonar (SSS). In this work, we apply deep learning to synthetic aperture sonar (SAS) imagery which has an advantage over traditional SSS in that SAS produces coherent high- and constant-resolution imagery. Despite the successes of deep learning, one drawback is the need for abundant labeled training data to enable success. Such abundant labeled data is not always available as in the case of SAS where collections are expensive and obtaining quality ground truth labels may require diver intervention. To overcome these challenges, we propose a domain-specific deep learning network architecture utilizing a unique property to complex-valued SAS imagery: the ability to resolve angle-of-arrival (AoA) of acoustic returns through k-space processing. By sweeping through consecutive incrementally advanced AoA bandpass filters, this technique generates a sequence of images emphasizing angle-dependent seafloor scattering and motion from biologics along the seafloor or in the water column.

15. SUBJECT TERMS
synthetic aperture sonar, machine learning, deep learning, frequency domain, k-space processing, phased array processing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	U	27	Isaac Gerg
u	u	u			19b. TELEPONE NUMBER (*Include area code*) 814-867-3598

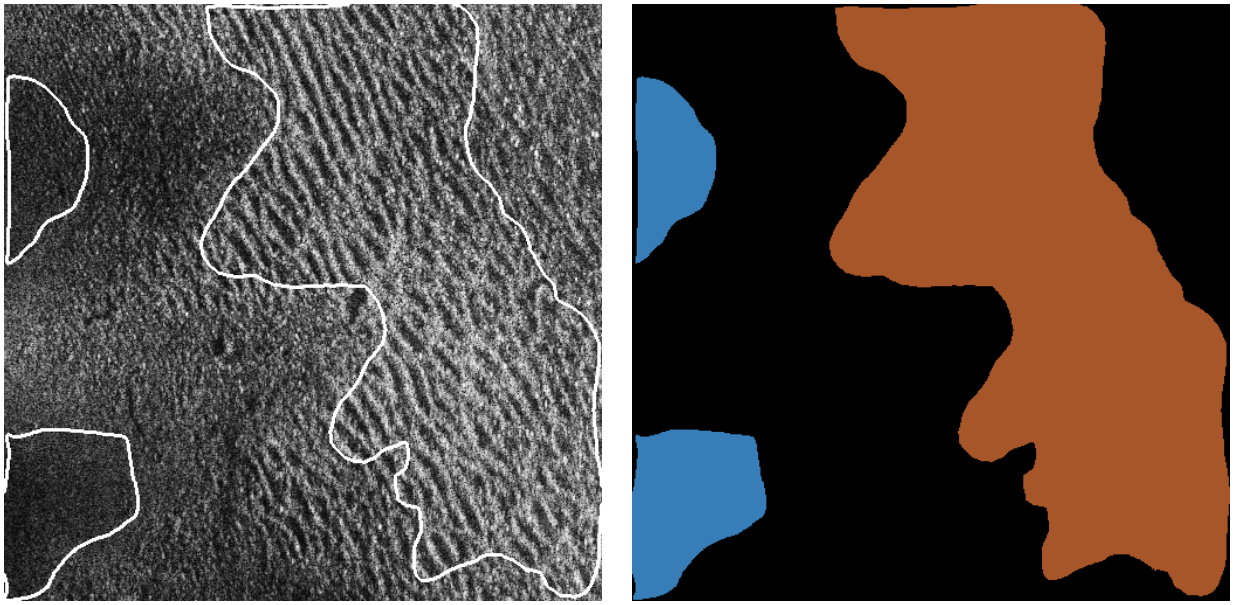


Figure 1: Example SAS image (left) with its weakly-labeled segmentation truth map (right). In the right image, the blue and brown areas indicate human-provided labels of two seafloor classes while the black area represents unlabeled pixels. The border of the labeled regions is shown in the left image in white. We see the human only labeled some of the "easy" portions of the image containing mostly unambiguous seafloor texture leaving the remaining pixels unlabeled.

coordinating divers to manually survey an area is time-consuming and logistically challenging. To mitigate this, researchers generally create labeled data post-survey using human experts who are accustomed to analyzing the imagery. However, this method presents its own challenges. Disambiguation of some seafloor textures (e.g., ripples [2]) requires imagery from multiple seafloor-to-sonar viewpoints since the same area of seafloor may appear different in a SAS image depending on the collection geometry.

Despite the aforementioned issues with obtaining labeled SAS imagery, we are given a *weakly-labeled* dataset where the data are labeled post-survey by a human, but only **some** of the "easy" portions of the image are labeled leaving the majority of the pixels unlabeled. Figure 1 shows an example weakly-labeled image from our dataset. Consequently, abundant SAS imagery may be collected during a survey, but due to the difficulty of the labeling process, labeled imagery remains scarce presenting a challenge to deep learning methods which usually require abundant labeled training data for success.

In this work, we utilize the fact that SAS imagery filtered intelligently in the k-space domain yields images with different "squints" or "looks" of the seafloor by steering the receive beam of the aperture. Sweeping over a consecutive set of look-angles creates a sequence of images akin to a movie calling attention to aspect-dependent scattering effects and motion of objects in the receive beam. Organizing such a process is the result of domain-knowledge of the SAS imaging modality and is sometimes called multi-look sequence processing (MLSP) from where our proposed network derives its name.

We make three main contributions with MLSP-Net:

1. **We introduce a novel domain-enriched network architecture utilizing magnitude and phase information contained in the complex-valued input SAS image to derive image features**

yielding improved performance over existing methods when labels and data are sparse as is often the case in SAS datasets. Abundant labels in SAS data are difficult to obtain because of the difficulty in determining accurate pixel-level labels. Our network architecture takes as input a single-look-complex SAS image atypical for current methods [3, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] which discard phase information and exclusively analyze the magnitude image. Our proposal seeks to utilize this often discarded data because of the rich information we know it contains from our domain-knowledge of the problem. We show our network outperforms many recent methods and especially yields improvement in discriminating between acoustic shadow and dark sand classes where the former is an artifact of the imaging geometry and the latter is a true seafloor type. Through an ablation study we show improvements in classification performance given by incorporation of the complex-valued image in a domain-relevant manner.

2. **We introduce a set of "filter modules" each ingesting the input complex-valued SAS image and use k-space filtering to generate a sequence of multi-look images based on acoustic angle-of-arrival (AoA) thus allowing our network to discover AoA dependent features.** Our proposed method explicitly models the temporal nature of this image sequence. Such an image sequence has been found to be useful for image interpretation by human analysts [14] and the authors' experience. Specifically, we model the multi-look sequence using a recurrent neural network (RNN) (i.e. Bidirectional 2D long-short-term-memory (LSTM)) enabling us to capture temporal correlations in the image sequence. The set of filter modules is initialized in an intelligent manner guided by domain-knowledge of the SAS imaging modality [14] thereby creating an image sequence emphasizing the aspect-dependent acoustic reflections of the seafloor as well as object movement in the acoustic path during the survey. As an example, schools of fish commonly appear as fuzzy clouds in the magnitude SAS image due to their movement during the formation of the synthetic aperture. Their presence obscures the seafloor resulting in segmentation difficulty, but through k-space filtering such occlusion may be mitigated (see Figure 2 for an example output). Likewise, classes such as shadow are easily confused with dark sand but become more discernible through multi-look sequence processing.

3. **We introduce a filter-kernel formulation for the "filter module" which is fully differentiable and therefore learnable as part of the training process.** The filter's structure takes on a frequency domain band-pass form derived from the k-space representation succinctly capturing acoustic angle-of-arrival information. Further, we devise a differentiable form of the filter so filter bandwidth, offset, and attenuation are all trainable parameters during the learning process allowing the network to be trained in an end-to-end fashion without the need to explicitly filter the images (using **fixed** filters) as a pre-processing step prior to the learning process.

APPROACH

Background

The output of SAS image reconstruction is an image containing complex-valued pixels called a single-look complex (SLC) image which encodes angle-of-arrival (AoA) of the scattered pulse in addition to its magnitude and phase. The 2D Fourier transform of the SLC yields a k-space representation with axes k_x and k_y for spatial dimensions of x and y for range and along-track respectively [15]. Once in k-space, interpolation is done to arrive at f, θ space where f is frequency and θ is look angle or AoA. For sufficiently low fractional bandwidth systems (i.e. f_s, the sampling rate of the system, is much smaller than f_0, the operating center frequency of the system), the angular

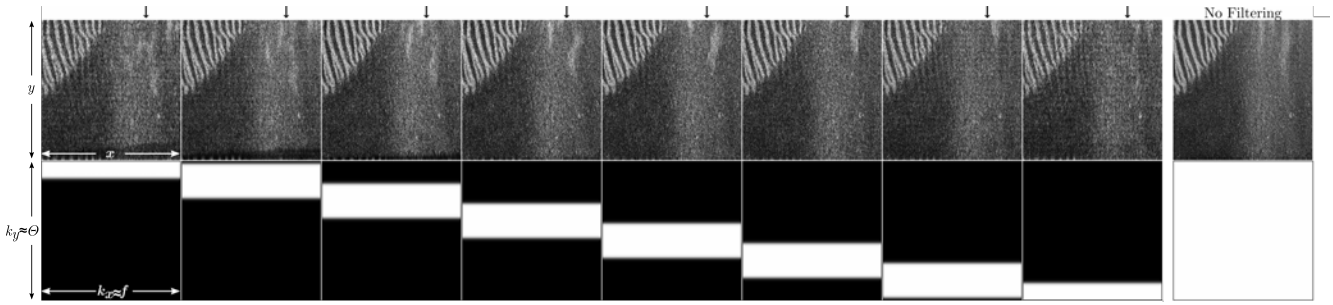


Figure 2: Example MLSP sequence showing benefits of k-space filtering a SAS image. (Top) Image squint sequence derived from SLC image (right-most column). (Bottom) Corresponding k-space filter with vertical axis approximately proportional to look-angle, θ, and horizontal axis approximately proportional to frequency. In the top right quadrant of the images of the top row, there is an arrow above each image pointing to a "fuzz" moving vertically in the image sequence. This is likely biologics in the water column moving during the collection of the synthetic aperture. The k-space filtering produces a pseudo-motion effect across frames of the image sequence capturing this movement. This phenomenon is not apparent when k-space filtering is disabled (see right-most column, "No Filtering" for this image) resulting in seafloor occlusion or false seafloor texture.

dependence on waveform arrival becomes independent of frequency. One can think of this in terms of the Fourier slice theorem [16, 17] when the ratio f_0/f_s is large enough, the arc formed by a small continuous section of look angles is locally linear (i.e. a small arc of a circle becomes linear as the circle diameter goes to infinity) resulting in lines of constant angle to be nearly horizontal (see Figure 4). Using this assumption, k_y is now approximately proportional to θ, the look angle. With this representation, band-pass filtering in θ across all frequencies and then performing an inverse 2D Fourier transform results in an SLC whereby the array is "squinted" and directed to a subset of look angles in θ; the beam geometry is shown in Figure 3. Squinting the receiver beam in this manner results in reduced azimuthal spatial resolution but at the benefit of disambiguating the AoA of the acoustic returns.

When we filter with a fixed beamwidth over consecutive angle offsets (Figure 2, bottom), we arrive at a sequence of images useful for image interpretation (Figure 2, top). Despite the reduced azimuthal resolution, we now discern aspect-dependent features such as those arising from acoustic shadows. In addition, we view the beam steering effect as a temporal filter whereby visible changes image-to-image in the sequence are the result of motion from objects (like fish or seagrass) during the formation of the synthetic aperture.

Justification of k_y as Angle-of-Arrival Approximation

To properly filter the AoA's in k-space we must know the sonar center frequency. It is not uncommon to be given SAS imagery with the sonar center frequency withheld by the manufacturer for proprietary reasons; this is the case for the data used in this work. Without the center frequency, we cannot exactly know the shape of the appropriate filter in θ (see Figure 4) but for systems with a sufficiently small fractional bandwidth (commonly referred to as high-frequency (HF) SAS systems), we can approximate the shape of the AoA filters which is what we do here. This section provides the mathematical model for this justification.

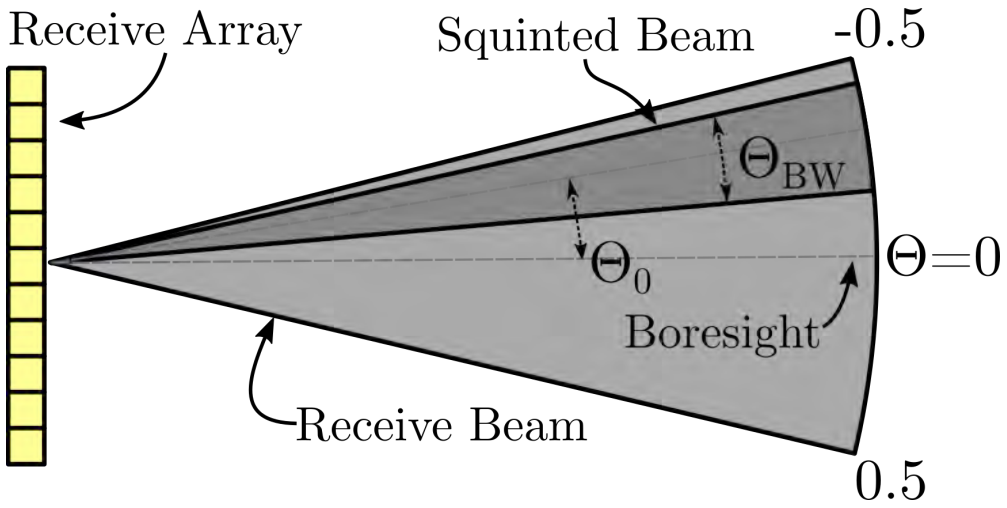


Figure 3: Model used for the parametric filter estimate. The yellow boxes represent the receiver array. θ_{BW} is squinted beam angle bandwidth and θ_0 is squinted beam angle offset. In addition to filter attenuation, these parameters are learned by MLSP-Net as part of the training procedure.

We begin by defining the k-space of an SLC as,

$$\mathbf{K} = \mathscr{F}^2\{\mathbf{x}\} \tag{1}$$

where $\mathscr{F}^2$ is the 2D Fourier transform of the input SLC, $\mathbf{x}$. $\mathbf{K}$ has axes of k_x, k_y. The k-space transform for a stripmap image (which is the imaging geometry of our data) is interpreted as roughly organizing incoming acoustic waves by AoA in the k_y dimension. Figure 4 depicts this intuition for the sample SAS given by [18] (recall we are not given the sonar center frequency hence our approximation). We see in the figure lines of constant AoA run approximately horizontally in k-space. If the sonar center frequency is known, we can perform a simple non-linear transform to go from k-space to $\theta - f$ space. The figure shows k-space with a colored overlay of isolines representing constant AoA with those of our approximation shown as the grey horizontal lines.

Using the geometry of the k-space representation, we approximate AoA as a scalar multiple α and k_y. Let's begin with the definition of θ from [15],

$$\theta = \arctan\left(\frac{k_y}{2k_0 + k_x}\right) \tag{2}$$

where k_0 is the carrier wavenumber (i.e., $\frac{2\pi f_0}{c}$, f_0 is the carrier frequency of the system, and c is the speed of sound in the medium). We seek to arrive at the form $\theta \approx \alpha k_y$ giving

$$\frac{\arctan\left(\frac{k_y}{2k_0+k_x}\right)}{k_y} \approx \alpha \tag{3}$$

and when k_0 is sufficiently large, the argument of the arctan is small giving us via $\arctan(x) \approx x$ since x is small,

$$\frac{\frac{k_y}{2k_0+k_x}}{k_y} \approx \alpha \tag{4}$$

Table 1: Fractional bandwidths for several SAS systems mentioned in the literature. We see for high-frequency SAS systems the maximum AoA error is less than five degrees.

Source	Sonar Type	Fractional Bandwidth
[19]	High-frequency	0.41
[20]	High-frequency	0.17
[18]	High-frequency	0.2
[20]	Low-frequency	0.5

$$\frac{1}{2k_0 + k_x} \approx \alpha \tag{5}$$

giving us our linear approximation,

$$\theta \approx \left(\frac{1}{2k_0 + k_x} \right) k_y \tag{6}$$

when the fractional bandwidth of the system is small enough.

Furthermore, we quantify the error of our approximation as a function of fractional bandwidth. If we are given a bin in k_y, we define the AoA error, ξ, as the difference between AoA at $f_{\min}$ and $f_{\max}$. For a given image resolution (Δ_x, Δ_y), f_0, and speed of sound (c), this error is defined as,

$$\xi = \theta_{\max} - \theta_{\min} = \arctan\left(\frac{\frac{\pi}{\Delta_y}}{2k_0 - \frac{\pi}{\Delta_x}} \right) - \arctan\left(\frac{\frac{\pi}{\Delta_y}}{2k_0 + \frac{\pi}{\Delta_x}} \right) \tag{7}$$

where ξ is the worst case AoA error for the given system parameters, and k_0 is the reference in k-space and defined as $k_0 = 2\pi \frac{f_0}{c}$. We see from Equation 7, as k_0 goes to infinity, the argument of the arctan goes to zeros thus making Equation 7 go to zero. Figure 5 shows ξ over a variety of fractional bandwidths, and in conjunction with Table 1, most high-frequency SAS systems commonly have $B_F \leq 0.4$ giving a maximum AoA error less than five degrees using the approximation of Equation 6, an acceptable error for the purposes of MLSP-Net. Finally, we see from Equation 7 lines of constant AoA become constant in k_y as the center frequency increases and the error of $\theta \approx \alpha k_y$ vanishes.

MLSP-Net Network Design & Data Flow

We first describe the overall flow of MLSP-Net and then describe each component in detail: the Multi-Look Sequence Path, the Static Image Path, and Final Image Segmentation. Figure 6 shows an architectural diagram of the method. Overall, the input to MLSP-Net is a complex-valued SLC image and the output is a vector of class probabilities for each pixel. Upon input, the SLC is in real-imaginary form with each pixel representing a complex value. The SLC is then fed to two paths, the Multi-Look Sequence Path (where temporal features are extracted) and the Static Image Path (where static features are extracted). The output of these paths is then concatenated and fed to a Final Image Segmentation network whose output is a vector class probabilities for each pixel.

For all convolutions in MLSP-Net, we use the "valid" padding scheme but prepend the convolutions with "reflect" padding along the height and width dimensions so the resultant convolution has the same height and width dimensions as the input. Unless otherwise specified, the network weights are initialized using a normal distribution scaled by the scheme in [21]. Tensorflow 2.5.0 [22] is used train our networks. All network parameters are updated every mini-batch via backpropagation since our

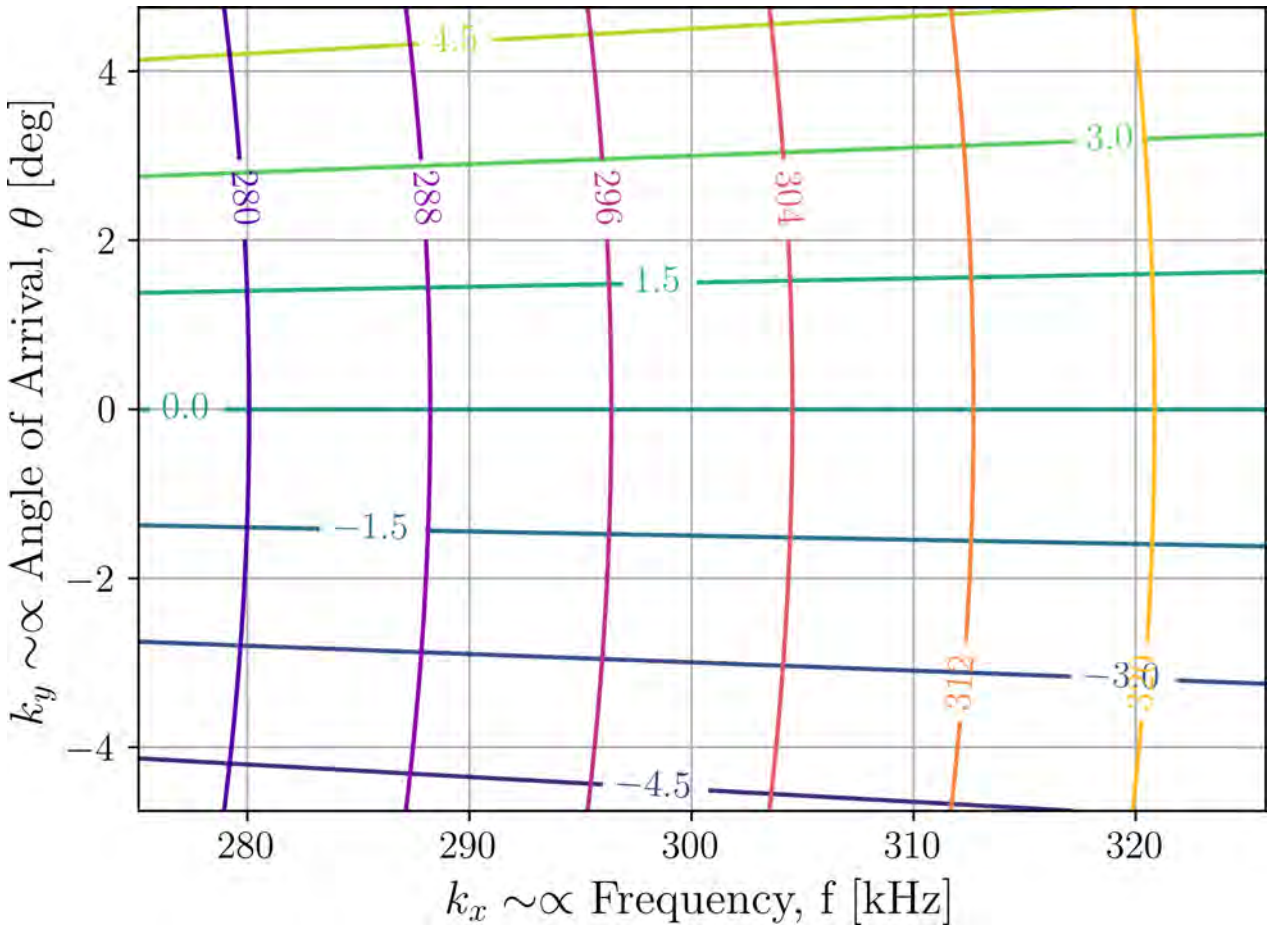


Figure 4: A depiction of the differences between true AoA with that of our approximation made in k-space for the SAS system of [18] (see 1) at 0.015m resolution. The picture overlays colored iso-lines of true constant AoA (horizontal lines) and frequency (vertical lines) onto k-space. The grey-colored grid attached to the axes k_x and k_y show the AoA and frequency approximation used in MLSP-Net (Equation 6) . Recall, we use this approximation because we are not given the sonar center frequency, f_0, for the data; not an uncommon scenario. We see no more than a degree of AoA error across the frequency band using our approximation.

network is fully differentiable from input to output. This includes all U-Nets, Filter-Modules, Squeeze and Excitation Modules, 2D Convolutional LSTM, and the Segmentation Net. MLSP-Net contains 5,103,159 trainable parameters.

Multi-Look Sequence Path

The Multi-Look Sequence Path begins with the input SLC and transforms it to k-space via a complex 2D Fourier transform (a differentiable operation). Next, the k-space map is copied eight times with each copy filtered by its respective Filter-Module numbered one through eight; this operation narrows the receive beam of each image to a subset of its full azimuthal beam pattern (see Figure 3 for a geometric diagram) and then steers it to the desired look-angle. The eight filtered images are then transformed back to the spatial domain via an inverse complex 2D Fourier transform (also a differentiable operation). Next, the magnitude of the resultant complex-valued images is taken and each image is dynamic range

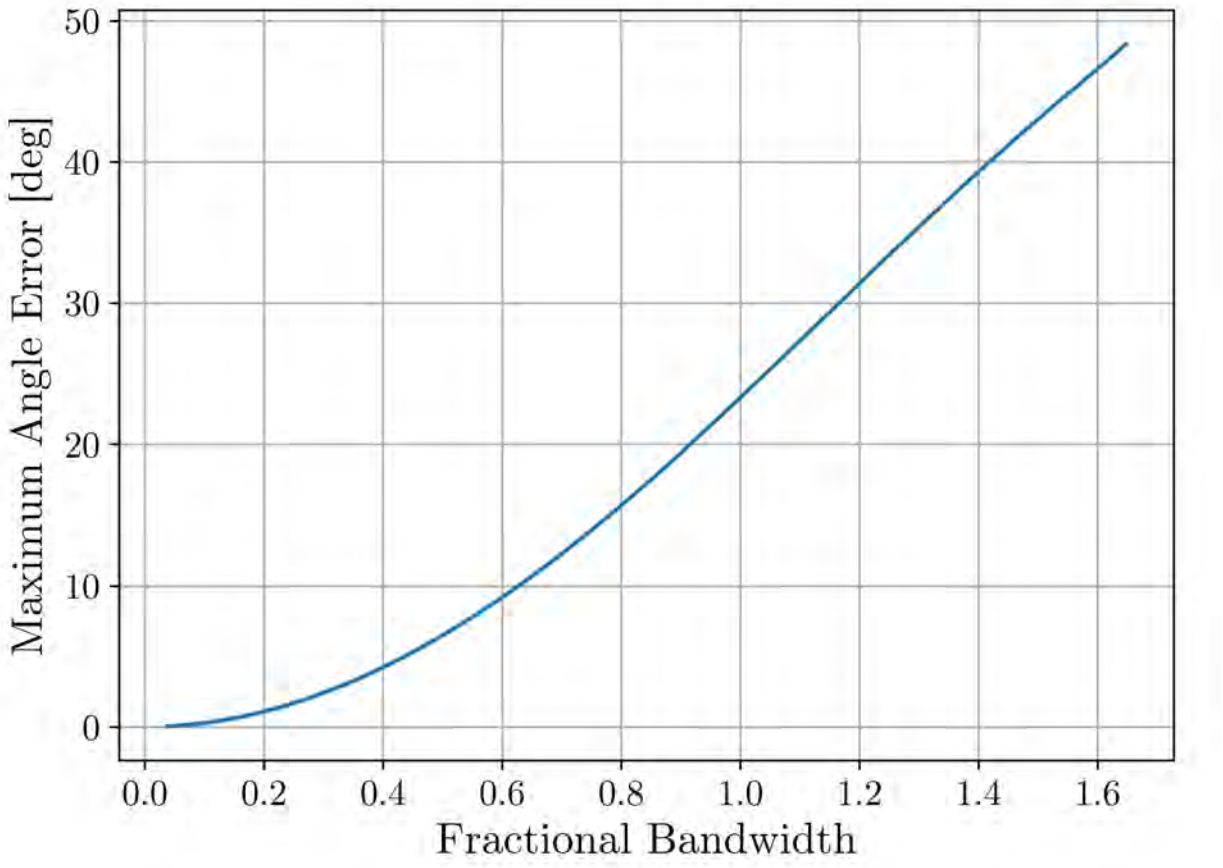


Figure 5: Maximum angle-of-arrival error as a function of fractional bandwidth. As the fractional bandwidth decreases, the angle-of-arrival approximation error decreases. Several high-frequency (HF) SAS systems (as used here) and their fractional bandwidths are in Table 1.

compressed (DRC'ed) using the differentiable method of [23]. Next, each image is downsampled by a factor of two in both dimensions using integration; this is done for memory and compute considerations. Next, each image is fed to a shared U-Net (see Table 5) whose output is fed to individual squeeze and excitation (S&E) modules [24] using a reduction ratio of $r = 8$. The final sequence of features maps are then fed to a bi-directional 2D convolutional long-term-short-term-memory (LSTM) model [25, 26]. 2D Convolutional LSTMs provide a mechanism to model spatial correlations over sequentially meaningful tensors by creating and updating a context vector after each tensor of the sequence is presented to the module. The bi-directional aspect of the module considers both forward and reverse instantiations of the sequence which we know from domain-knowledge are equally relevant. The 2D convolutional LSTM consists of sixteen kernels each of size 5×5. The resultant output represents the end of the Multi-Look Sequence Path and has shape $256 \times 256 \times 32$.

Each Filter-Module contains a differentiable filter function specified in the k-space domain which is capable of learning and fine-tuning the filter parameters during the training procedure. We accomplish this by constructing the filter to be of a band-pass form defined by a rational approximation of the

rectangle function shown in Equation 8 as,

$$\text{rect}(t) = \lim_{n\to\infty, n\in\mathbb{Z}} \frac{1}{(2t)^{2n}+1} \tag{8}$$

where t is the domain. Figure 7 shows the formation of the rectangle shape of the filter as n increases. Consequently, defining the rectangle function as a soft approximation using a fixed $n \ll \infty$ yields a smooth function alleviating Gibbs artifacts normally present when using a hard rectangle filter. Furthermore, we specify the relevant filter parameters of bandwidth, offset, and attenuation all in a differentiable manner as given in Equations 9 and 10,

$$\text{softrect}_n(t) = \frac{1}{(2t)^{2n}+1} \tag{9}$$

$$\text{softfilter}(\theta, \theta_0, \theta_{\text{BW}}, \alpha) = \text{softrect}_n\left(\frac{\theta-\theta_0}{\theta_{\text{BW}}}\right)\cdot\alpha + (1-\alpha) \tag{10}$$

where n controls the filter fidelity (i.e., the "sharpness" of the filter), $\theta_0 \in [-0.5, 0.5]$ is the filter offset, $\theta_{\text{BW}} \in [\varepsilon, 1]$ is normalized look angle bandwidth, and $\alpha \in [\varepsilon, 1]$ is the attenuation. Figure 8 illustrates the how the filter is defined.

We now address the initialization of the filter parameters for the Filter-Modules. Recall Figure 2 whereby the consecutive sequence of images filtered by incremental AoA advances yields a richer semantic interpretation of the images than the "No Filtered" image alone. Moreover, the sequence loses some of its interpretability if it is randomly shuffled. Therefore, we initialize the k-space filters in a domain-enriched fashion to initially produce a useful ordered temporal sequence based on author experience and [14]. The resulting initialization yields a sequence that is productively used by the 2D convolutional LSTM. Conversely, choosing the k-space filter parameters (e.g., $\theta_{\text{BW}}, \theta_0, \alpha$) randomly may not induce this meaningful image sequence we wish our network to exploit. We initialize the k-space filters in the Filter-Modules as follows: (1) the positions (e.g., θ_0) are spaced uniformly across look angle with a bandwidth of $\theta_{\text{BW, i}} = 0.6$, and (2), the attenuation is set to $\alpha_i = 0.5$. Finally, we use a filter fidelity of $n = 10$ (reference Equation 9). Consequently, for a given set of input parameters $\{\theta_{0,i}, \theta_{\text{BW},i}, \alpha_i\}$, the Filter-Module, i, the 2D k-space filter is constructed as

$$\text{Filter-Module}_i = \text{softfilter}(\theta, \theta_{0,i}, \theta_{\text{BW},i}, \alpha_i) \otimes \mathbf{1}^T \tag{11}$$

where $\theta = \{r : r = \frac{n}{255} - 0.5, n \in \{0, 1, ...255\}\}$ and $\otimes$ is the Kronecker product (used as a broadcasting operator here).

Static Image Path

The Static Image path processes the input SLC similar to existing SAS segmentation methods by examining the magnitude SLC image.

This path ingests the input SLC and downsamples it by a factor of two in each dimension using averaging. Next, we convert the SLC to a magnitude image and then standardize it using

$$\text{Standardize}(x) = \frac{|x| - \text{mean}(|x|)}{\text{stddev}(|x|)} \tag{12}$$

where x is the input SLC and stddev is the standard deviation. Through cross-validation, we found the best results by standardizing the image as opposed to applying DRC as done in the Multi-Look

Table 2: Pixel accuracy by class and mean pixel accuracy (MPA) for each method. Larger numbers indicate better performance. Classes are listed in Table 3. Best for each class in bold. We see the MLSP-Net has the best MPA of all the methods and yields the best results for four of the seven classes. Moreover, we show improved performance over the next best method for the shadow class, which is easily confused with dark sand (MPA of 0.632 for MLSP-Net-No-Filter versus 0.714 for MLSP-Net), demonstrating the utility of the Filter-Modules and 2D modeling of the MLSP features.

Method	Pixel Accuracy							MPA
	SW	SD	SL	SG	RK	RS	RL	
Lianantonakis, *et al.* [29]	0.00	0.64	0.46	0.17	0.67	0.69	0.79	0.49
Williams [8]	0.72	0.04	0.84	0.23	0.85	0.25	0.97	0.56
Zare, *et al.* [30]	**1.00**	0.00	0.21	0.54	0.62	0.00	0.01	0.34
Rahnemoonfar, et al. [31]	0.477	0.971	0.971	0.811	**0.972**	0.719	0.997	0.845
Sun, *et al.* [32]	0.81	0.87	0.87	0.69	0.96	0.45	0.98	0.80
MLSP-Net-No-Filter	0.632	**0.986**	0.969	**0.831**	0.951	0.974	0.997	0.906
MLSP-Net	0.714	0.977	**0.978**	0.816	**0.972**	**0.986**	**0.998**	**0.920**

Sequence Path. This image is then input to a U-Net (see Table 5) which is a separate U-Net from the Multi-Look Sequence Path. The resultant output represents the end of the Static Image Path and has the shape $256 \times 256 \times 16$.

Final Image Segmentation

The two resultant feature maps from each path (Multi-Look Sequence Path and Static Image Path) are concatenated in the feature dimension and then passed to a "Segmentation Net" (see Table 4).

Segmentation Net contains a "Global Channel Weight (GCW)" layer, 2D spatial dropout [27] with dropout proportion set to 0.8, and finally a 1x1 "2D" Convolution [28]. GCW functions similarly to a squeeze and excitation (S&E) network[24] but we remove the dependency on the input data so the same weighting is given to each input sample. Through cross-validation, we found this re-weighting scheme gives us better results than a canonical S&E scheme. Our re-weighting scheme is input-independent, unlike S&E, which we think benefits our limited training data scenario in preventing overfitting with input-dependent weighting. In this way, we get the good benefits of channel re-weighting as given by S&E networks but forgo making the re-weighting input-dependent. We initialize all the channel weights to unity at the start of training. The output of Segmentation Net is a set of logits for each pixel.

The output of Segmentation Net is upsampled by a factor of two using nearest-neighbor interpolation. Finally, a softmax operation is performed along the logits dimension to arrive at $\hat{\mathbf{y}}$, the predicted class probabilities for each pixel. The resultant output represents the end of Final Image Segmentation and the output of MLSP-Net. The final output shape is $512 \times 512 \times 7$.

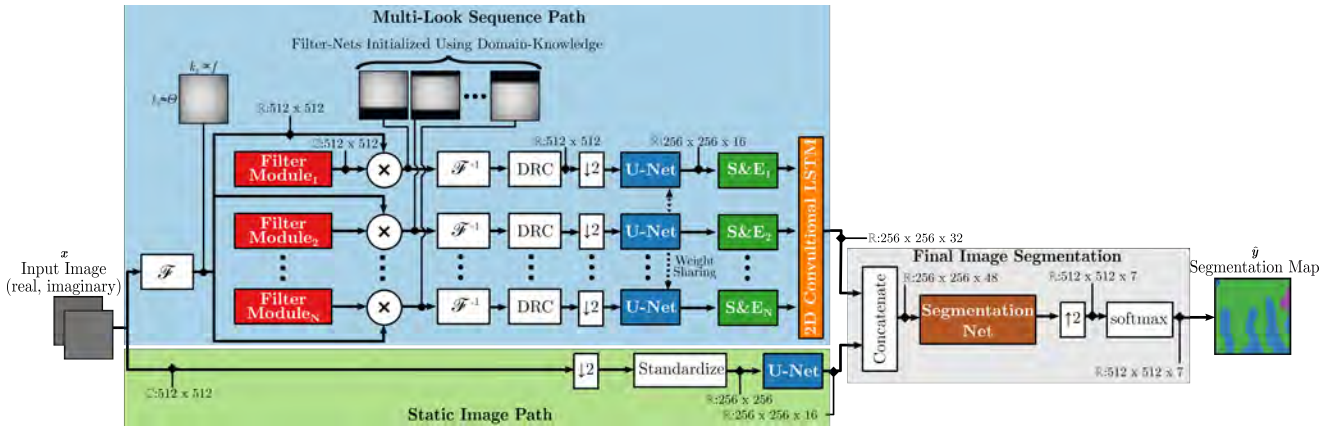


Figure 6: Our proposed network called multi-look sequence processing network (MLSP-Net) for SAS segmentation is composed of two primary paths: Multi-Look Sequence and Static Image. The Multi-Look Sequence path is composed of eight Filter-Modules operating in k-space and are initialized using domain knowledge so an initial useful multi-look sequence emerges (see Figure 2). The eight looks are sent through a shared U-Net encoder, re-weighted by squeeze and excitation modules, and then fed to a bidirectional 2D convolutional LSTM to extract meaningful features from the image sequence. The Static Image path simply ingests the input image and processes it through a U-Net. Features from both paths are then concatenated in the feature dimension and processed by a segmentation network for Final Image Segmentation. We show the utility of the MLSP by doing an ablation study whereby we initialize and fix the Filter-Modules to unity thereby removing their effect. Now, all eight paths have the same non-filtered input image resulting in the same feature map for each step of the LSTM. Results in Table 2 show the benefit of the MLSP than without it. Circular taps in the figure show representative results; diamond taps show data type (real $\mathbb{R}$ or complex $\mathbb{C}$) and tensor shape.

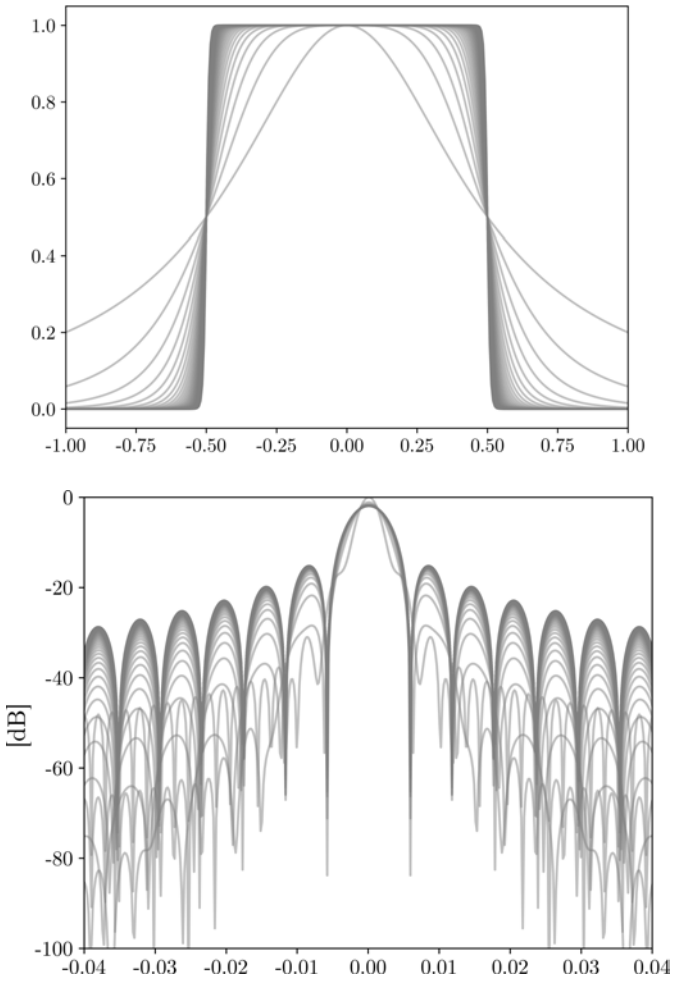


Figure 7: (Top) Convergence to the rectangle function by the limit of a rational function in Equation 8. The figure shows a growing sequence of n ultimately converging to the rectangle function as $n \to \infty$. (Bottom) Magnitude Fourier transform (i.e., magnitude impulse response) of top showing convergence to a sinc function as $n \to \infty$. The parameter n in Equation 8 allows us to trade off filter fidelity with ringing artifacts of the impulse response (i.e., Gibbs phenomenon) as well as dampening the coefficients away from the origin to effectively reduce the support of the filter. We use $n = 10$ for each Filter-Module in MLSP-Net.

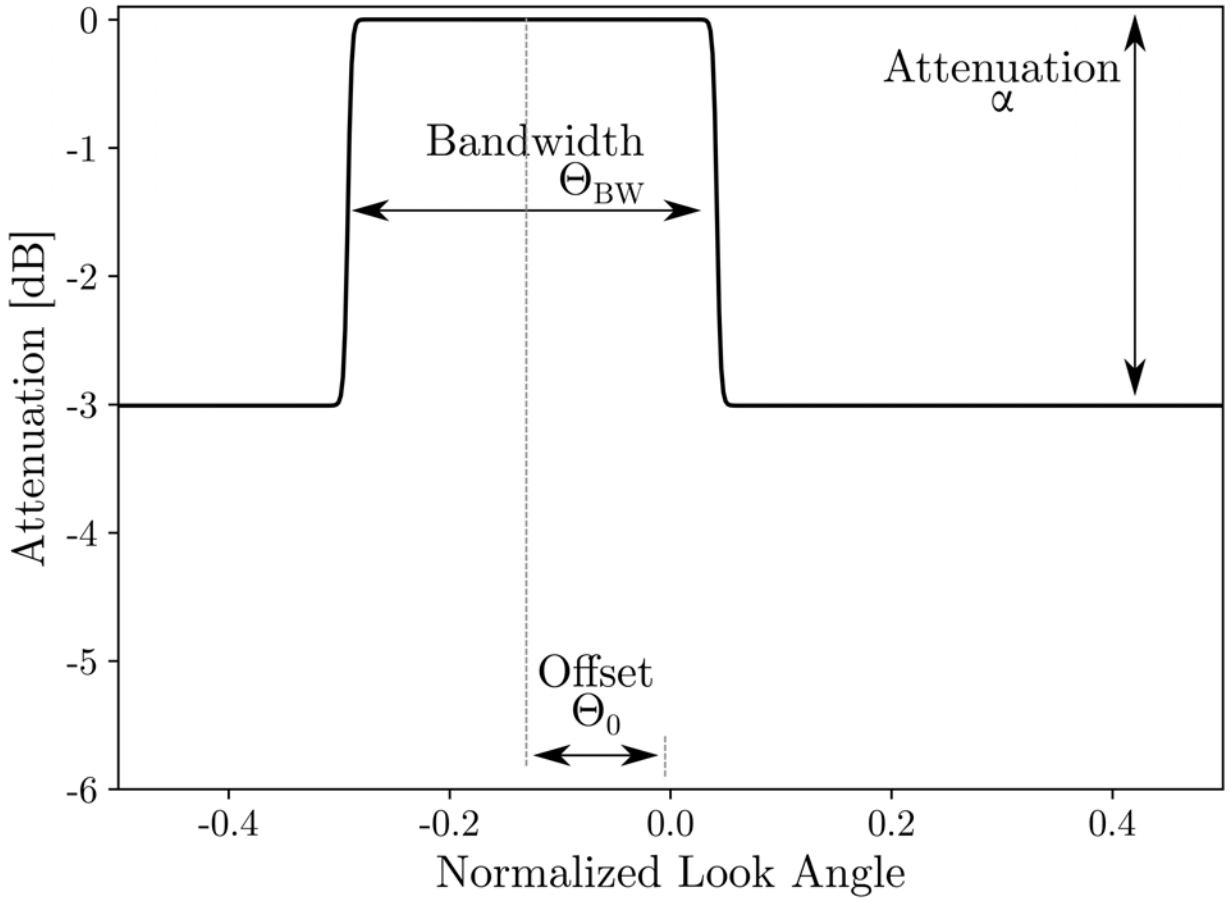


Figure 8: A sample output of softfilter $_{10}(\theta, \theta_{BW} = \frac{1}{3}, \theta_0 = -0.125, \alpha = 0.5)$ ***using*** $n = 10$ ***in Equation 9. The softfilter*** $_{10}$ ***function is used in each Filter-Module of Figure 6 to filter the input SLC in the*** k***-space domain to filter by specific look-angles of acoustic energy. See Figure 2 for an example of these filters applied to a SAS image and Figure 4 to see*** k***-space coordinates of an example high-frequency SAS system which the softfilter*** $_{10}$ ***function is applied across all frequencies to filter by look-angle.***

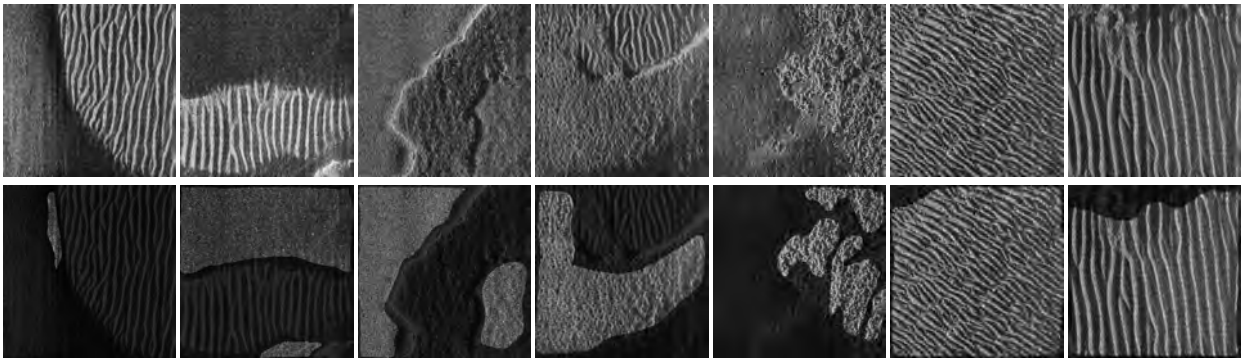


Figure 9: Seafloor classes considered in this work using data from [33]. For each image pair (i.e., column), the image on the top is the original SAS image and the image on the bottom highlights one of the corresponding seafloor classes present in the top image deemed "easy" to label by a human. In viewing (a) and (b) we see classes like dark sand and shadow look similar in magnitude imagery and thus are easily confused. However, we resolve this ambiguity by examining angle-of-arrival information through multi-look processing employed by our proposed network architecture, MLSP-Net.

Table 3: Classes making up our dataset along with the corresponding colors used in plots.

Class Name	Abbreviation	Color
Shadow	SW	Red
Sand (dark)	SD	Blue
Sand (light)	SL	Green
Seagrass	SG	Purple
Cobble / Rock	RK	Yellow
Ripple (Small)	RS	Brown
Ripple (Large)	RL	Pink

WORK COMPLETED

We have implemented the mask framework using the k-priors/modeling we proposed in to a new network we call MLSP-Net, tested said network on a real-world SAS dataset, compared results to contemporary SAS segmentation algorithms, and published the results in the IEEE International Geoscience and Remote Sensing Symposium and have a paper accepted with minor revisions to IEEE Transactions on Geoscience and Remote Sensing.

RESULTS

Dataset Description

The imagery used to evaluate our experiment is from a high-frequency SAS aboard an unmanned underwater vehicle [33]. Each image starts as size 1001×1001 pixels and is downsampled using integration to 512×512 pixels for training/evaluation due to memory constraints. The dataset totals 113 SLC images composed of seven labeled classes (Table 3 and Figure 9) and one "unlabeled" class. No location information or other metadata was provided for the images including the sonar center frequency, f_0. The sizes of the training and test sets are 80 and 33 images respectively. Of the training data, we use 64 images for training and 16 images for validation. The distribution of labels is class imbalanced and shown in Figure 11.

Experimental Setup

We trained MLSP-Net for 500 epochs using Adam [34] with a learning rate of 10^{-3} and a mini-batch size of 64. Training a single model of MLSP-Net takes approximately 6.5 wall clock hours on an NVIDIA Titan X Pascal GPU. The training iteration giving the best validation result is used to evaluate the test set. The loss function used to train the network is the weighted categorical focal loss [35] given by Equation 13,

$$\mathscr{L} = \sum_{k=1}^{h} \sum_{kk=1}^{w} \mathbf{w}_{\mathbf{y}}(k,kk) \cdot \mathscr{L}_{\text{focal}}(\mathbf{y}(k,kk), \hat{\mathbf{y}}(k,kk)) \tag{13}$$

where h, w are the image height and width respectively, $\mathbf{w}_{\mathbf{y}} \in \mathbb{R}^{h \times w}$ is an indicator map of $\{0,1\}$ noting if the pixel is labeled (i.e., see Figure 1 right; we do no accrue loss for unlabeled pixels), $\mathscr{L}_{\text{focal}}$ is the categorical focal loss [35], $\mathbf{y} \in \mathbb{R}^{h \times w \times 7}$ are the ground truth labels for each pixel, and $\hat{\mathbf{y}} \in \mathbb{R}^{h \times w \times 7}$ is the estimated labels for each pixel.

We evaluate and compare methods using mean pixel accuracy (MPA) because of its robustness to class

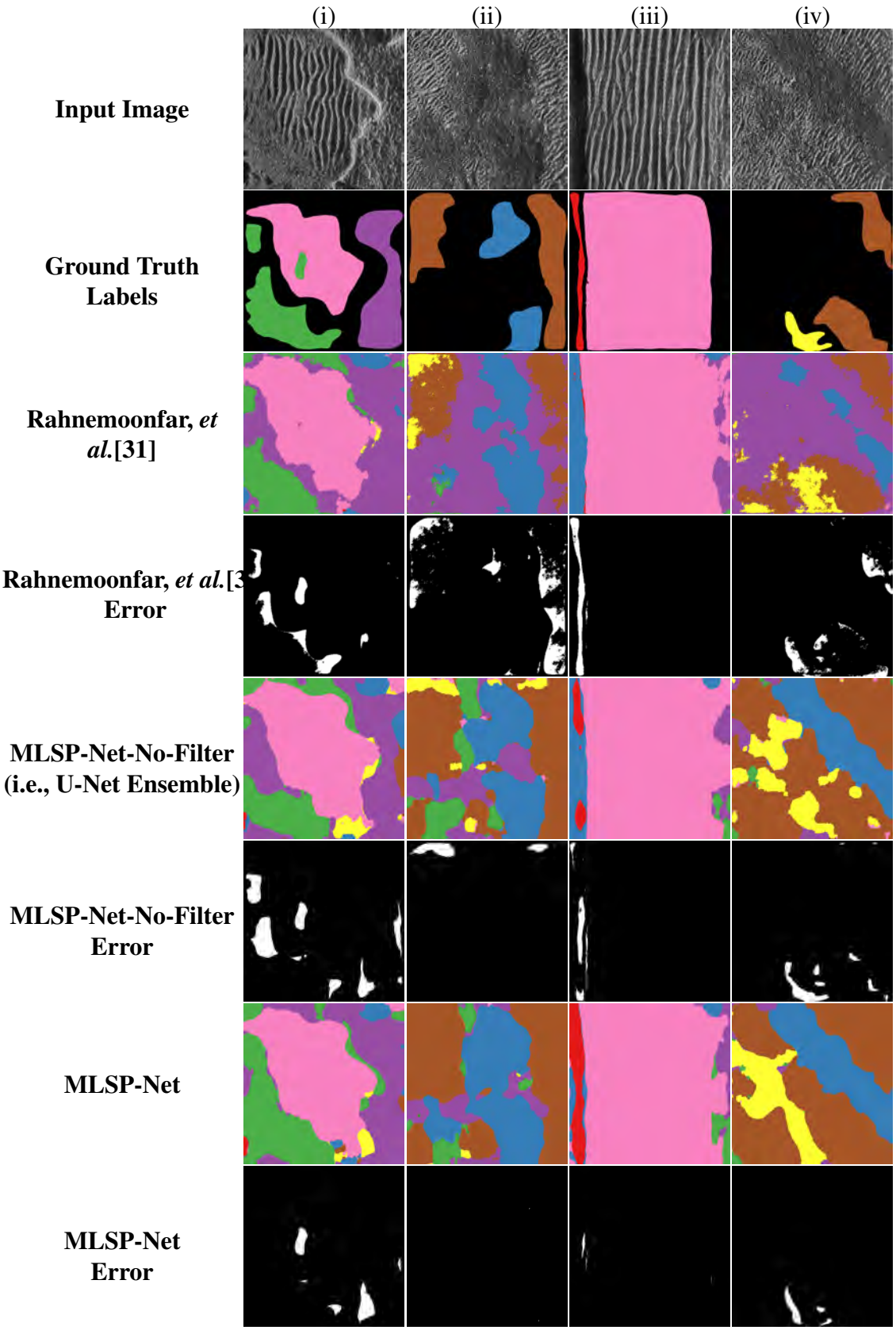


Figure 10: *Sample results from the top three methods of Table 2: Rahnemoonfar, et al.[31]; our proposed method with ablation MLSP-Net-No-Filter (which is effectively a U-Net ensemble); and our full proposal MLSP-Net. For rows top to bottom: input image; ground truth labels (recall these are weak/partial labels); Rahnemoonfar, et al.[31] results; classification errors between ground truth and Rahnemoonfar, et al.[31]; MLSP-Net-No-Filter results; classification errors between ground truth and MLSP-Net-No-Filter, MLSP-Net results; and classification errors between ground truth and MLSP-Net. As shown, MLSP-Net makes fewer classification errors overall compared to the other methods. Furthermore, MSLP-Net classifies the shadow area on the left side of the image (iii) better than our proposed method with ablation (MLSP-Net-No-Filter) thus demonstrating the improvement given by the AoA filtering. See Table 3 for class-color mappings.*

Table 4: Description of the Segmentation Net block of Figure 6.

Layer Name	Layer Function	Dim.	# Filters	Input
gcw	Global Channel Weighting	N/A	N/A	N/A
s2dd	Spatial 2D Dropout (0.8)	N/A	N/A	gcw
output	2D Conv	1 x 1	7	s2dd

Table 5: Description of the U-Net model based around a pre-trained Resnet50 encoder [36]. For conciseness, we list the decoder side of the network. The U-Net multiplies the input by 128 to be consistent with the scaling used during the pre-training. Upsampling uses the nearest neighbor method.

Layer Name	Layer Function	Dim.	# Filters	Input
conv1a	Pad+Conv+BN+GELU	3x3	128	conv3_block4_out
conv1b	Pad+Conv+BN+GELU	3x3	128	conv1a
up2	Upsampling	2x2	N/A	conv1b
conv2a	Pad+Conv+BN+GELU	3x3	64	up2, conv2_block3_out
conv2b	Pad+Conv+BN+GELU	3x3	64	conv2a
up3	Upsampling	2x2	N/A	conv2b
conv3a	Pad+Conv+BN+GELU	3x3	32	up3, conv1_relu
conv3b	Pad+Conv+BN+GELU	3x3	32	conv3a
up4	Upsampling	2x2	N/A	conv3b
conv4a	Pad+Conv+BN+GELU	3x3	16	up4
output	Pad+Conv+BN+GELU	3x3	16	conv4a

imbalance which is present in our dataset. We define MPA as,

$$\text{MPA} = \frac{1}{7}\sum_{c=1}^{7} \text{Acc}_c \tag{14}$$

where Acc_c is the accuracy of class c defined in a one-versus-all manner as,

$$\text{Acc}_c = \frac{\text{TP}_c + \text{TN}_c}{\text{total number of labeled pixels for class c}} \tag{15}$$

where TP_c represents number of true positives in a one-vs-all scenario and is defined as the total number of labeled pixels where $\arg\max \mathbf{y}(x,y) = c$ and $\arg\max \hat{\mathbf{y}}(x,y) = c$, and TN_c represents the number of true negatives in a one-vs-all scenario and is defined as the total number of labeled pixels where $\arg\max \mathbf{y}(x,y) \neq c$ and $\arg\max \hat{\mathbf{y}}(x,y) \neq c$. Note, the set of $\text{Acc}_c, c \in \{1,2,...6,7\}$ are the diagonal elements of the confusion matrix (see Figure 12) and MPA is the mean of the diagonal elements. Note, the denominator of Equation 15 only considers the number of ground-truth **labeled** pixels across all images since this is a weakly-labeled dataset with less than 45% of pixels labeled in the test set (see Figure 11).

Comparison with State-of-the-Art Algorithms

We compare our results with three state-of-the-art methods used specifically in sonar/SAS seabed environment image segmentation: Lianantonakis, *et al.* (2007), Williams (2009) [8], and Zare, *et al.* (2017) [30]. We also compare against a SOTA deep learning segmentation algorithm, the U-Net [1] [37].

[1]There are many well-known deep learning segmentation algorithms, but they are not designed to incorporate the SAS-

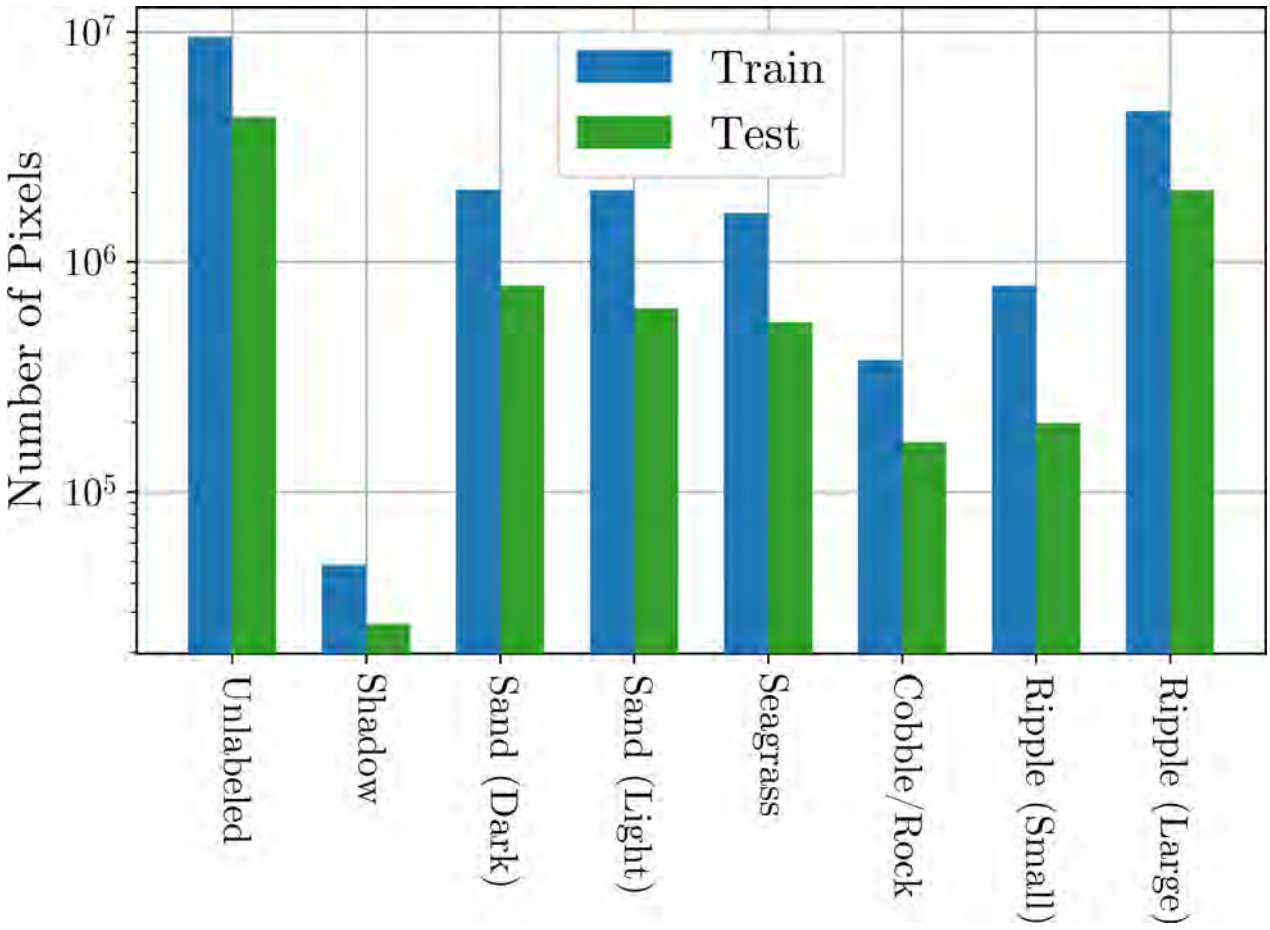


Figure 11: Distribution of each class (including unlabeled pixels) for the training and test sets. The dataset was weakly labeled hence a large number of unlabeled pixels. Moreover, the dataset is highly class imbalanced.

In this section, we give the implementation details we use in generating the comparison methods as no source code is publicly available to evaluate. We make a best effort attempt to reproduce the methods as given in their respective sources. Our implementation is based on Python 3.7 code running on an Intel (R) Core (TM) i9-7960X 2.80GHz CPU with Linux operating system.

Lianantonakis, et al. (2007) [29]. This method uses Haralick features [38] derived from the gray-level co-occurrence matrix (GLCM) and couples this with active contours to arrive at a binary class mapping. We extend this work to multiple classes by simply using the same feature descriptors as the original work but apply k-means++ [39] to cluster; a similar replication approach is used in [6]. We ran k-means++ with one-hundred random initializations and selected the run producing the minimum within-cluster sum of squares error in a manner consistent with [8].

Williams (2009) [8]. This method uses wavelet features along with spectral clustering to compute the segmentation map. We found spectral clustering results in similar performance as using k-means++ so we opt to use for simplicity as we did in Lianantonakis, *et al.* mentioned above; a similar replication

specific insight of MLSP. MLSP-Net can use any SOTA segmentation module. However, MLSP-Net provides SAS-specific enhancements over black-box methods such as a U-Net.

approach is used in [6].

Zare, et al. (2017) [30]. In this work, the feature sets are produced by Sobel edge descriptors (Sobel) [40], histograms of oriented gradients (HOG) [41], and local binary pattern (LBP) features [42]. For each feature descriptor, we use the same sliding window strategy of Lianantonakis, *et al.*[29] to derive a feature vector for each pixel.

Rahnemoonfar, et al. (2019) [31]. In this work, a deep network composed of dilated convolutions, dense modules, and inception modules is used to perform semantic segmentation for the automatic extraction of potholes in SSS imagery. We compare against this method despite being applied to SSS imagery because it was recently developed, supervised (few supervised algorithms exist for SAS segmentation), and obtains SOTA results on a real SSS dataset. We train this algorithm using the Adam optimizer and a learning rate of 10^{-3} as specified by the paper. We select the epoch for the test set evaluation using the same scheme as MLSP-Net which is to use early stopping based on the MPA of the validation set.

Sun, et al. (2021) [32]. This work combines a super-pixel method with deep learning using an algorithm similar to Deep Cluster [43] to cluster the data and assign class labels. Each image is parsed by a deep network and then super-pixels are formed from the pixel embedding and updated periodically during training. The network is trained by iteratively learning the pseudo-labels generated by the super-pixels and the super-pixel assignment is updated every 200 iterations.

We report confusion matrices and mean pixel accuracy (MPA) to assess performance. MPA is more robust to class imbalance so we use it as a metric rather than traditional pixel accuracy. Notably, we cannot apply the commonly used segmentation metric of intersection over union (IoU) because only **some** portion of each class is labeled and the true class probability for a pixel may be a mixture of classes (i.e., mixture of seafloor textures). Consequently, these properties make the intersection and union operations of IoU not applicable to this setup.

Table 2 shows the pixel average per class and the mean pixel average (MPA) for several comparison methods. We see in the Table our MLSP-Net method has the highest MPA of all the methods and provides best classification for classes light sand (SL), rock (RK), small ripple (RS), and large ripple (RL). Additionally, we see superior performance over MLSP-Net-No-Filter, essentially a U-Net, for the shadow class which is especially easy to confuse with the dark sand class demonstrating the benefit of activating the learnable Filter-Module. Figure 12 shows the confusion matrices for the top two methods Table 2 of ten runs of MLSP-Net and MLSP-Net-No-Filter. The Figure shows MLSP-Net has better performance on average for the majority of classes but also shows reduced variance demonstrating its robustness to training sample selection.

Ablation Study

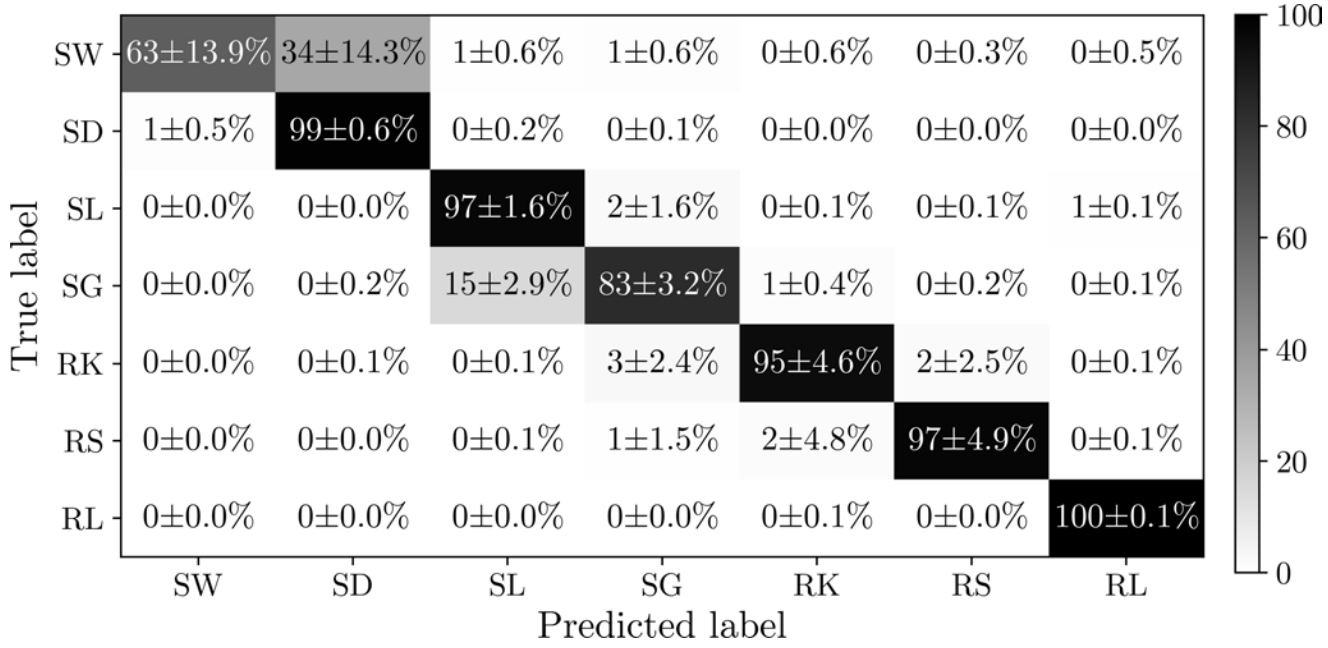
To demonstrate the necessity of the AoA filters in MLSP, we train MLSP-Net without this processing step and report results. This is accomplished by nulling the filtering operation in each Filter-Module in Figure 6 so no filtering occurs and all images fed to the convolutional 2D LSTM module are exactly the same thus removing any notion of a temporal sequence. Mathematically, this is accomplished by making the parameters of the Filter-Modules fixed and setting $\theta_{0,i} = 0, \theta_{\mathrm{BW},i} = 1.0$, and $\alpha_i = 0$ in Equation 10. We refer to this configuration as "MLSP-Net-No-Filter" in our results. This configuration essentially results in two U-Nets in a small ensemble with magnitude-only input imagery so we forgo explicitly evaluating a U-net for comparison since it is already represented by this configuration.

Figure 13 shows a distribution of MPA over ten runs between our proposed method and

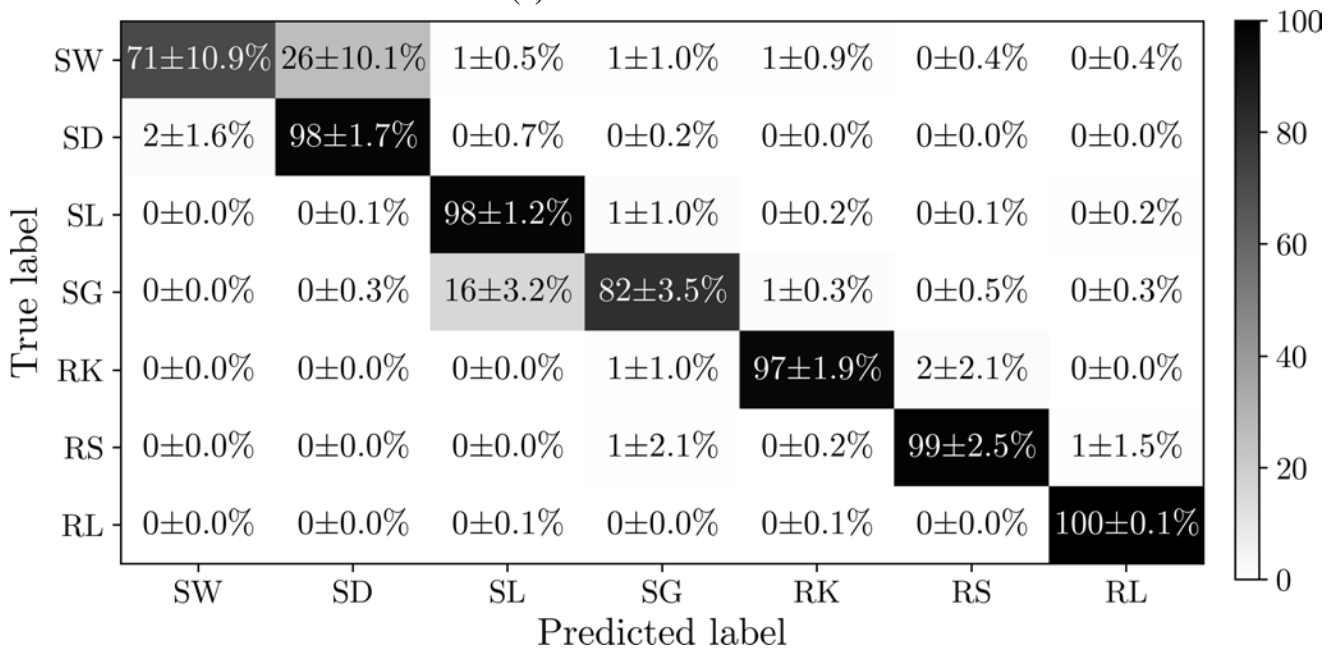
MLSP-Net-No-Filter. The figure shows our proposed method has a higher median (0.921 versus 0.902) and mean MPA (0.920 versus 0.906) than the next best method, MLSP-Net-No-Filter. Finally, Figure 10 shows example segmentations of our proposed method and the next two best methods, MLSP-Net-No-Filter and Rahnemoonfar, *et al.*[31].

Filter Parameters During Training

We show each Filter-Module's bandwidth, attenuation, and position as a function of training epoch for one run of MLSP-Net in Figure 14. We see from the figure the most significant changes occur with filter position and bandwidth and the least change occurs with filter attenuation. Figure 14.b shows a thirteen percent increase in filter bandwidth of a near-boresight filter denoted in red for each subfigure indicating the network emphasizing boresight AoA features.



(a) MLSP-Net-No-Filter



(b) MLSP-Net

Figure 12: Confusion matrices for the two best methods of Table 2, our proposed MLSP-Net method and baseline MLSP-Net-No-Filter. Class abbreviations defined in Table 3. These results are composed of ten training runs. We see MLSP-Net yields better MPA in four of the seven classes and does especially better at differentiating shadow and dark sand classes. Note, the MLSP-Net-No-Filter configuration is essentially two U-Nets put together in a small ensemble so we forgo explicitly evaluating the U-Net as a comparison since it would be redundant.

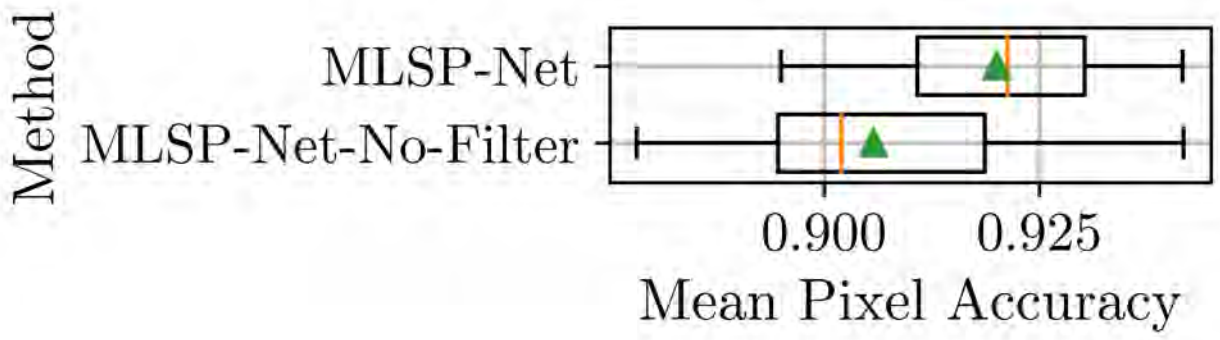
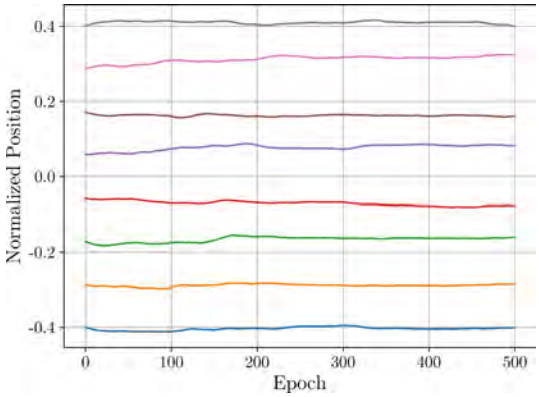
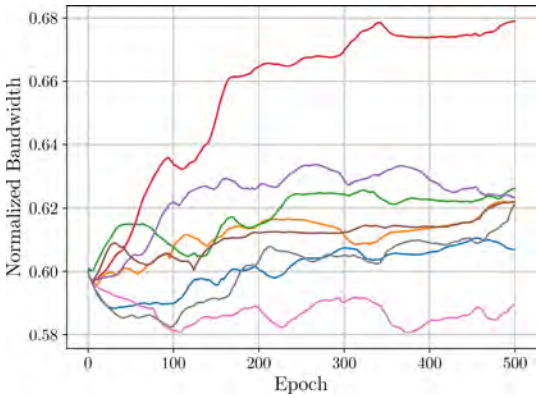


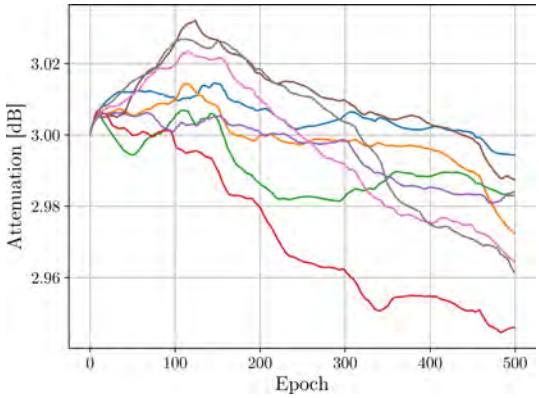
Figure 13: Box and whisker plot of MPA over ten runs of the two best methods of Table 2. Orange lines of the boxplot indicate median values (0.921 versus 0.902) and green triangles indicate average (0.920 versus 0.906). We see the MLSP-Net yields better performance and less sensitivity to training sample selection for training than MLSP-Net-No-Filter which does no k-space filtering. Thus, this test demonstrates not only the utility of this domain-knowledge in improving network performance but also its robustness to training sample selection which is especially important when abundant data is not available as is the case here.



(a) Positions



(b) Bandwidths



(c) Attenuations

Figure 14: The eight Filter-Module parameters as a function of training epoch for one run of MLSP-Net. We see the most significant changes occur in filter bandwidth and position where the filter near boresight is allocated the largest bandwidth.

IMPACT/APPLICATIONS

We hope this work motivates researchers to consider the complex-valued SAS image (i.e., SLC) as input to deep learning algorithms in the future. Traditionally, phase information has been discarded as most SAS machine learning algorithms focus their efforts exclusively on magnitude imagery. Our results demonstrate: (1) improvements in image segmentation performance by exploiting the complex-valued nature (i.e., phase) of the SAS SLC, (2) a network design exploiting phase information using traditional signal processing techniques in a differentiable manner, and (3) reduced algorithm complexity by enabling end-to-end training thus forgoing the need to pre-process the SLC into magnitude imagery prior to network input.

***Publications from this Program:**

- I. D. Gerg and V. Monga, "Synthetic Aperture Sonar Image Segmentation Using Adaptive, Learned Beam Steering," IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium, 2022, pp. 983-986, doi: 10.1109/IGARSS46834.2022.9883235.
- I. D. Gerg and V. Monga, "Deep Multi-Look Sequence Processing for Synthetic Aperture Sonar Image Segmentation," provisional acceptance with minor revisions to IEEE Transactions on Geoscience and Remote Sensing, Oct 2022.

TRANSITIONS

This work constitutes basic research. Transitions to prototype and/or production systems are future directions.

RELATED PROJECTS

No Related Projects

REFERENCES

[1] O. Bryan, R. E. Hansen, T. S. Haines, N. Warakagoda, and A. Hunter, "Challenges of labelling unknown seabed munition dumpsites from acoustic and optical surveys: A case study at Skagerrak," *Remote Sensing*, vol. 14, no. 11, p. 2619, 2022.

[2] D. P. Williams and E. Coiras, "On sand ripple detection in synthetic aperture sonar imagery," in *IEEE ICASSP*. IEEE, 2010, pp. 1074–1077.

[3] S.-M. Steele, J. Ejdrygiewicz, and J. Dillon, "Automated synthetic aperture sonar image segmentation using spatially coherent clustering." *JOT*, vol. 16, no. 3, 2021.

[4] J. T. Cobb and A. Zare, "Boundary detection and superpixel formation in synthetic aperture sonar imagery," in *IOA SAS/SAR*, Sept. 2014.

[5] J. T. Cobb and J. Principe, "Seabed segmentation in synthetic aperture sonar images," in *DSMEOOT*, vol. 8017. ISOP, 2011, p. 80170M.

[6] ——, "Autocorrelation features for synthetic aperture sonar image seabed segmentation," in *ICSMC*. IEEE, 2011, pp. 3341–3346.

[7] J. T. Cobb and A. Zare, "Multi-image texton selection for sonar image seabed co-segmentation," in *DSMEOOT*, vol. 8709. ISOP, 2013, p. 87090H.

[8] D. P. Williams, "Unsupervised seabed segmentation of synthetic aperture sonar imagery via wavelet features and spectral clustering," in *ICIP*. IEEE, 2009, pp. 557–560.

[9] ——, "Fast unsupervised seafloor characterization in sonar imagery using lacunarity," *TGRS*, vol. 53, no. 11, pp. 6022–6034, 2015.

[10] A. Zare, N. Young, D. Suen, T. Nabelek, A. Galusha, and J. Keller, "Possibilistic fuzzy local information C-means for sonar image segmentation," in *SSCI*. IEEE, 2017, pp. 1–8.

[11] J. Peeples, D. Suen, A. Zare, and J. Keller, "Possibilistic fuzzy local information C-means with automated feature selection for seafloor segmentation," in *DSMEOOT*, vol. 10628. ISOP, 2018, p. 1062812.

[12] D. Kohntopp, B. Lehmann, D. Kraus, and A. Birk, "Seafloor classification for mine countermeasures operations using synthetic aperture sonar images," in *OCEANS*. IEEE, 2017, pp. 1–5.

[13] B. Gips, "Bayesian seafloor characterization from SAS imagery," in *UACE*, 2019.

[14] D. Cook, R. Hansen, A. Lyons, and A. Yezzi, "Motion tracking of transient refractive effects in SAS imagery using optical flow," in *IOA SAS/SAR*, 2014.

[15] D. W. Hawkins, "Synthetic aperture imaging algorithms: with application to wide bandwidth sonar," Ph.D. dissertation, University of Canterbury. Electrical and Computer Engineering, 1996.

[16] M. Soumekh, *Synthetic aperture radar signal processing*. New York: Wiley, 1999, vol. 7.

[17] H. J. Callow, "Signal processing for synthetic aperture sonar image enhancement," Ph.D. dissertation, University of Canterbury. Electrical and Electronic Engineering, 2003.

[18] A. Bellettini and M. Pinto, "Design and experimental results of a 300-khz synthetic aperture sonar optimized for shallow-water operations," *JOE*, vol. 34, no. 3, pp. 285–293, 2009.

[19] D. Billon and F. Fohanno, "Theoretical performance and experimental results for synthetic aperture sonar self-calibration," in *IEEE OCEANS*, vol. 2, 1998, pp. 965–970 vol.2.

[20] G. S. Sammelmann, J. E. Fernandez, J. T. Christoff, L. Vaizer, J. D. Lathrop, R. W. Sheriff, and T. C. Montgomery, "High-frequency/low-frequency synthetic aperture sonar," in *DRTMMT II*, A. C. Dubey and R. L. Barnard, Eds., vol. 3079, ISOP. SPIE, 1997, pp. 160 – 171.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*. IEEE, 2015, pp. 1026–1034.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[23] I. D. Gerg, D. C. Brown, S. G. Wagner, D. Cook, B. N. O'Donnell, T. Benson, and T. C. Montgomery, "GPU acceleration for synthetic aperture sonar image reconstruction," in *IEEE OCEANS*, 2020, pp. 1–9.

[24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*. IEEE, 2018, pp. 7132–7141.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *ANIPS*, vol. 28, 2015.

[27] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *CVPR*. IEEE, 2015, pp. 648–656.

[28] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[29] M. Lianantonakis and Y. R. Petillot, "Sidescan sonar segmentation using texture descriptors and active contours," *JOE*, vol. 32, no. 3, pp. 744–752, 2007.

[30] A. Zare, N. Young, D. Suen, T. Nabelek, A. Galusha, and J. Keller, "Possibilistic fuzzy local information C-Means for sonar image segmentation," in *SSCI*. IEEE, 2017, pp. 1–8.

[31] M. Rahnemoonfar and D. Dobbs, "Semantic segmentation of underwater sonar imagery with deep learning," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 9455–9458.

[32] Y.-C. Sun, I. D. Gerg, and V. Monga, "Iterative, deep, and unsupervised synthetic aperture sonar image segmentation," in *OCEANS*. IEEE, 2021, pp. 1–5.

[33] J. T. Cobb, "Synthetic aperture sonar seabed environment dataset (SASSED)," 2018. [Online]. Available: https://data.mendeley.com/datasets/s5j5gzr2vc/3

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*. IEEE, 2017, pp. 2980–2988.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*. IEEE, 2016, pp. 770–778.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *ICMICCAI*. Springer, 2015, pp. 234–241.

[38] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *TSMC*, no. 6, pp. 610–621, 1973.

[39] D. Arthur and S. Vassilvitskii, "K-Means++: The advantages of careful seeding," in *Symposium on Discrete Algorithms*. ACM-SIAM, 2007, p. 1027–1035.

[40] H. Frigui and P. Gader, "Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic *k*-nearest neighbor classifier," *TFS*, vol. 17, no. 1, pp. 185–199, 2009.

[41] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[42] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *TIP*, vol. 19, no. 6, pp. 1657–1663, 2010.

[43] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *ECCV*, 2018, pp. 132–149.