STARS: Anomaly Guidance for Zero Shot Sim-to-Real Transfer for Segmentation in Sonar Imagery (Supplementary Information)

Advaith Venkatramanan Sethuraman
advaiths@umich.edu
Katherine A. Skinner
kskin@umich.edu

Department of Robotics
University of Michigan
Ann Arbor, Michigan, USA

For more information about STARS and the associated side scan sonar dataset: https://umfieldrobotics.github.io/STARS.github.io/.

1 STARS Implementation Details

Although a side scan sonar image is gray-scale, we repeat it to produce an input image with 3 channels. This reduces any modification to backbone networks. Our network uses ResNet-34 backbones with ImageNet pretrained weights for initialization [5]. The teacher's weights (α_t) are frozen for both training and inference. The architecture of ϕ follows that of a UNet with depth 4 ($D_l = 4$)[9]. Detailed per-layer information with dimensions can be seen in Table (1).

For training, we use the AdamW optimizer with $lr = 1e^{-4}$ and batch size of 2. We train the network for 183 epochs, which takes 17 hours on a single Nvidia A40 GPU. The network was trained to convergence based on the validation IOU on the synthetic dataset. None of the networks see a single real shipwreck image during training. RandAug was used as data augmentation [3]. Networks are trained and evaluated with image resolution of $(1728, 1728)$.

A sigmoid activation is applied to the output of the last layer and thresholded at 0.5 to produce a binary mask. All UpSampling/DownSampling operations are bilinear. The DoubleConv layer is defined as $Conv2d \rightarrow BatchNorm2d \rightarrow ReLU \rightarrow Conv2d \rightarrow BatchNorm2d \rightarrow ReLU$. STARS is implemented in Pytorch.

1.1 Student/Teacher Training

The student and teacher are two distinct networks and the weights are not shared. It has been found that pretrained image features learned from large datasets like ImageNet are useful for anomaly detection [8]. Inspired by these findings, we use a ResNet-34 pretrained on ImageNet for both the teacher and student encoder. The training process is as follows: the teacher encoder is always frozen during training and does not receive gradient updates. The student encoder is not frozen and receives weight updates.

Table 1: Detailed per-layer information for STARS. STARS uses ResNet34 backbones for its encoders. $\oplus$ denotes concatenation. Tensors indicated with * are used *only* for training, and are not required for inference. Synthetic sonar image S and real general terrain image T are used for training, but inference takes only real shipwreck image R as input.

Input Name	**Input Description**	**Input Dimension**
S*	Synthetic Side Scan Image	$3 \times 1728 \times 1728$
T*	Real Side Scan General Terrain Image	$3 \times 1728 \times 1728$
R	Real Side Scan Shipwreck Image	$3 \times 1728 \times 1728$
Tensor Name	**Layer Description**	**Out. Dimension**
Student Encoder α_s (ResNet 34 Backbone)		
#0	Conv2d(DownSample(S))	$64 \times 128 \times 128$
#1	ResNet Layer(#0) Feature	$64 \times 64 \times 64$
#2	ResNet Layer(#1) Feature	$128 \times 32 \times 32$
#3	ResNet Layer(#2) Feature	$256 \times 16 \times 16$
#4	ResNet Layer(#3) Feature	$512 \times 8 \times 8$
Student Decoder β_s		
#5	ConvTranspose2d(#4) $\oplus$ #3 $\rightarrow$ DoubleConv	$256 \times 16 \times 16$
#6	ConvTranspose2d(#5) $\oplus$ #2 $\rightarrow$ DoubleConv	$128 \times 32 \times 32$
#7	ConvTranspose2d(#6) $\oplus$ #1 $\rightarrow$ DoubleConv	$64 \times 64 \times 64$
$\hat{T}_p(1)$	GlobalAveragePool(UpSample(#7))	$64 \times 1 \times 1$
$\hat{T}_p(2)$	GlobalAveragePool(UpSample(#6))	$128 \times 1 \times 1$
$\hat{T}_p(3)$	GlobalAveragePool(UpSample(#5))	$256 \times 1 \times 1$
$\hat{T}_p(4)$	GlobalAveragePool(UpSample(#4))	$512 \times 1 \times 1$
Teacher Encoder α_t (ResNet 34 Backbone)		
#12	Conv2d(DownSample(T))	$64 \times 128 \times 128$
#13	ResNet Layer(#12) Terrain Feature	$64 \times 64 \times 64$
#14	ResNet Layer(#13) Terrain Feature	$128 \times 32 \times 32$
#15	ResNet Layer(#14) Terrain Feature	$256 \times 16 \times 16$
#16	ResNet Layer(#15) Terrain Feature	$512 \times 8 \times 8$
#17	Conv2d(DownSample(S))	$64 \times 128 \times 128$
$\tilde{f}_t(1)$	ResNet Layer(#17) Teacher Feature	$64 \times 64 \times 64$
$\tilde{f}_t(2)$	ResNet Layer($\tilde{f}_t(1)$ Teacher Feature	$128 \times 32 \times 32$
$\tilde{f}_t(3)$	ResNet Layer($\tilde{f}_t(2)$ Teacher Feature	$256 \times 16 \times 16$
$\tilde{f}_t(4)$	ResNet Layer($\tilde{f}_t(3)$ Teacher Feature	$512 \times 8 \times 8$
$T_p(1)$*	GlobalAveragePool(#13)	$64 \times 1 \times 1$
$T_p(2)$*	GlobalAveragePool(#14)	$128 \times 1 \times 1$
$T_p(3)$*	GlobalAveragePool(#15)	$256 \times 1 \times 1$
$T_p(4)$*	GlobalAveragePool(#16)	$512 \times 1 \times 1$
A	UpSample(#18) $\oplus$ UpSample(#19) $\oplus$ UpSample(#20) $\oplus$ UpSample(#21)	$960 \times 64 \times 64$
$A(1)$	CosDistance($\hat{T}_p(1)$, $\tilde{f}_t(1)$)	$64 \times 64 \times 64$
$A(2)$	CosDistance($\hat{T}_p(2)$, $\tilde{f}_t(2)$)	$128 \times 32 \times 32$
$A(3)$	CosDistance($\hat{T}_p(3)$, $\tilde{f}_t(3)$)	$256 \times 16 \times 16$
$A(4)$	CosDistance($\hat{T}_p(4)$, $\tilde{f}_t(4)$)	$512 \times 8 \times 8$
Deformation Network ϕ (ResNet 34 Backbone)		
#23	Conv2d(DownSample(S))	$64 \times 128 \times 128$
#24	ResNet Layer(#23) Feature	$64 \times 64 \times 64$
#25	ResNet Layer(#24) Feature	$128 \times 32 \times 32$
#26	ResNet Layer(#25) Feature	$256 \times 16 \times 16$
#27	ResNet Layer(#26) Feature	$512 \times 8 \times 8$
#28	ConvTranspose2d(#27 $\oplus$ A(4)) $\oplus$ #26 $\oplus$ A(3) $\rightarrow$ DoubleConv	$256 \times 16 \times 16$
#29	ConvTranspose2d(#28) $\oplus$ #25 $\oplus$ A(2) $\rightarrow$ DoubleConv	$128 \times 32 \times 32$
#30	ConvTranspose2d(#29) $\oplus$ #24 $\oplus$ A(1) $\rightarrow$ DoubleConv	$64 \times 64 \times 64$
#31	ConvTranspose2d(#30) $\oplus$ #23 $\rightarrow$ DoubleConv	$32 \times 128 \times 128$
#32	ConvTranspose2d(#31) $\oplus$ DownSample(S) $\rightarrow$ DoubleConv	$16 \times 256 \times 256$
$\hat{D}$	UpSample(Conv2d(#32))	$30 \times 1728 \times 1728$
Segmentation Decoder β_{seg}		
$\hat{M}$	Conv2d(UpSample(A) $\oplus\hat{D}$)	$1 \times 1728 \times 1728$

2 Baseline Details

We wish to emphasize that there are no baselines in the space of zero shot sim-to-real transfer for side scan sonar imagery. Instead, we choose a variety of baselines across computer vision that respect out data restrictions. First, we consider the SOTA anomaly detection method PatchCore since it is a valid approach when access to real examples is restricted [10]. Next, we consider domain adaptation approaches since they explicitly adapt features learned during training to a different domain for inference [4, 6]. We choose a SOTA segmentation model HRNetv2 as our naive baseline as it uses no sim-to-real transfer [11]. We also evaluate the performance of side scan sonar specific segmentation models (Yang et. al [12] and Burguera and Bonin-Font [1]). We include a salient object detection baseline [7] as it may have different inductive biases than the segmentation baseline HRNetv2. All baselines were trained with recommended hyperparameters to ensure fair comparison.

2.1 Baseline Implementation Details

- **PatchCore** is an unsupervised anomaly detection algorithm that only has access to real terrain images without any shipwrecks [10]. Although PatchCore was designed for industrial anomaly detection, we assert that no such state-of-the-art anomaly detection method exists specifically for side scan sonar imagery. We use PatchCore as a general proxy of SOTA anomaly detection methods for images.
- **HRDA** is a state of the art unsupervised domain adaptation technique [6]. We used the provided MiT Segformer pretrained weights as initialization.
- **PODA** is a state of the art zero shot unsupervised domain adaptation technique [4]. The target domain text prompt we use is “real side scan sonar imagery". We use a DeepLabv3 network with ResNet101 backbone with ImageNet pretrained weights [2, 5]
- **HRNetV2+OCR** is a state of the art semantic segmentation network [11, 13]. This serves to demonstrate segmentation performance when training on simulated data and testing on real data with no sim-to-real transfer. We use the HRNetV2-W48+OCR ImageNet pretrained weights as initialization.
- **Yang et. al** is a multi-channel semantic segmentation network for side scan sonar imagery [12].
- **Burguera and Bonin-Font** is a convolutional encoder/decoder network for semantic segmentation of side scan sonar imagery [1].
- **InSPyReNet** is a state of the art Salient Object Detection network [7]. Since our task is binary segmentation, we wish to compare to salient object detection methods that specialize in separating an object from the scene. We use the provided InSPyReNet_SwinB_HU weights trained on high resolution images for initialization.

2.2 Burguera Baseline Performance

We evaluated both Yang et. al [12] and Burguera and Bonin-Font [1] as side scan sonar specific segmentation methods. We report the best performing baseline (Yang et. al) in the main paper but report the performance of Burguera and Bonin-Font under the same training conditions in Table (2).

Table 2: Performance of Burguera segmentation baseline compared to all other baselines and STARS

Method	IOU_{ship} ↑	IOU_{terr} ↑	mIOU ↑	F1 Score ↑
PatchCore [■]	0.28	0.91	0.60	0.43
HRDA [■]	0.19	0.97	0.58	0.29
PODA [■]	0.28	0.97	0.63	0.41
HRNetV2 [■]	0.35	0.97	0.66	0.48
Yang et. al [■]	0.31	**0.98**	0.65	0.48
Burguera [■]	0.25	0.97	0.61	0.38
InSPyReNet [■]	0.33	0.97	0.65	0.45
STARS (ours)	**0.42**	**0.98**	**0.70**	**0.55**

3 Additional Experiments

3.1 Detailed Per-Site Results

Table 3: Per-site IOU_{ship} segmentation performance trained on **only** simulated data. Last column is averaged across all 14 sites.

Site	Barge #1	E.B.A	Flint	Grecian	Haltiner	H.F.	Lucy	Monrovia	Reef	Rend	Thew	Viator	Wilson	Montana	IOU_{ship} ↑
PatchCore [■]	0.30	0.36	0.33	0.21	0.20	0.03	0.40	0.25	0.01	0.22	0.45	0.17	0.53	0.51	0.28
HRDA [■]	0.32	0.34	0.11	0.51	**0.25**	0.03	0.04	0.40	0.00	0.09	0.05	0.29	0.13	0.09	0.19
PODA [■]	0.11	0.41	0.38	0.14	0.01	**0.07**	0.37	0.20	0.02	**0.36**	0.37	0.27	0.66	0.61	0.28
HRNetV2 [■]	**0.45**	0.50	0.44	0.53	0.08	0.00	0.42	**0.54**	0.03	0.11	0.27	0.33	0.74	0.47	0.35
Yang et. al [■]	0.22	0.40	0.42	0.45	0.15	0.04	0.22	0.38	**0.06**	0.22	0.33	0.33	0.63	0.51	0.31
Burguera [■]	0.12	0.33	0.30	0.47	0.03	0.02	0.30	0.34	0.02	0.16	0.28	0.29	0.48	0.43	0.25
InSPyReNet [■]	0.13	**0.62**	0.30	**0.73**	0.01	0.01	0.27	0.37	0.01	0.36	0.16	**0.49**	0.63	0.52	0.33
STARS (ours)	**0.45**	0.51	**0.52**	0.50	0.14	0.02	**0.60**	0.36	0.02	0.35	**0.56**	0.40	**0.79**	**0.64**	**0.42**

We present detailed per-site segmentation results. STARS meets or surpasses the performance of baselines consistently and maintains the highest IOU_{ship} averaged across all 14 sites. H.F. stands for Heart Failure. These unique failure cases motivate future work to develop methods that can perform sim-to-real transfer for extremely unstructured scenes.

3.2 Varying Real Terrain Dataset Size

Table 4: Segmentation Performance vs. Number of Real Terrain Images (N_T). ±1 standard deviation, N=5.

N_T	IOU_{ship} ↑	F1 Score ↑
10	0.35±0.03	0.49±0.04
50	0.38±0.03	0.53± 0.03
150	0.39±0.01	0.54± 0.02
250	0.40±0.01	0.54±0.01
312	0.41±0.01	0.54± 0.01

When deploying an AUV in a new body of water, it would be ideal to perform preliminary surveys and collect real terrain data for training. Then, the AUV can be deployed with a trained network better suited to the local terrain. We investigated how the amount of real terrain data affects the performance of our model by varying the number of randomly chosen real terrain images during synthetic data generation $N_T \in \{10, 50, 150, 250, 312\}$. We train 5 models per N_T and report the mean and standard deviation of IOU_{ship} in Table (4). We find that increasing the amount of real data does increase performance of the model, but only to a certain extent. Remarkably, at **only 10 real images** the model achieves 0.35 IOU_{ship} (still outperforming the best baseline HRNet with access to 312 real terrain images) and increasing N_T provides diminishing returns.

3.3 STARS Qualitative Results

We present larger, higher resolution network predictions illustrating the effectiveness of STARS in the shipwreck segmentation task in Figure (1). Note a failure case at the *Reef* site, where STARS mistakenly labels distractor objects (shaped like shipwrecks) as shipwrecks.

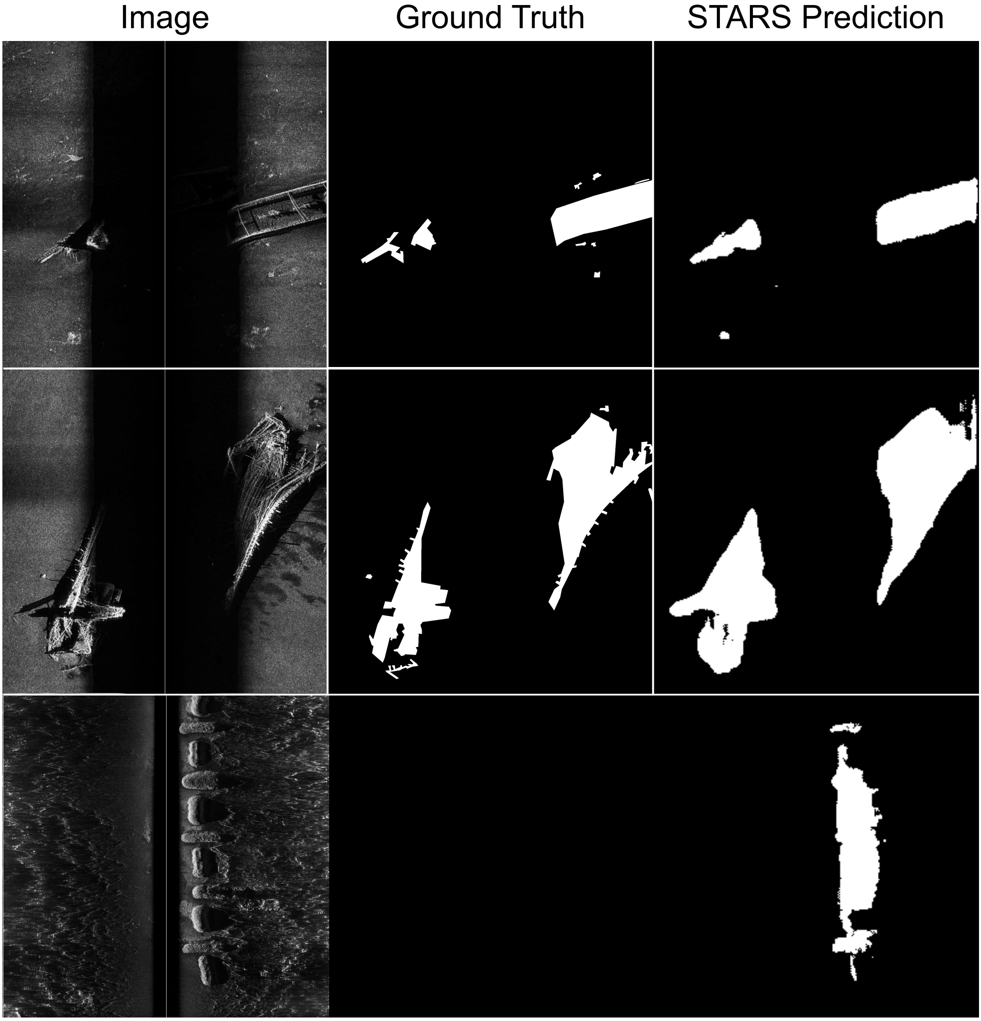

Figure 1: (top to bottom) Shipwreck sites pictured: Wilson, Montana, Reef. Zoom in for more detail. STARS has not seen a real shipwreck during training and performs zero shot sim-to-real transfer from synthetic images.

3.4 Comparison to Supervised Segmentation on Real Data

We wish to investigate the performance of a model when trained on all the real data we have access to. We fine-tune an HRNetV2+OCR model on real, labeled data from our field work. We call this model *HRNetV2 w/ Real Supervision*. We train 14 models, repeatedly withholding one site as a test site and training/validating on an 80/20 split of the remaining images. This model achieved an IOU_{ship} of 0.38. The detailed per-site results are shown in Table (5). We found that when unique sites like *Rend, Thew, Barge #1* are used as test sites, the other training sites are not representative of the test site. This can cause decreased performance. Training on real data does have value: *HRNetV2 w/ Real Supervision* performs better than HRNetV2 trained only on synthetic ships (0.35 IOU_{ship}) in Table (3). The fact that our method (**0.42** IOU_{ship}) outperforms *HRNetV2 w/ Real Supervision* can be attributed to the deformation and anomaly prediction networks, not the addition of synthetic data, as shown in Ablation Studies.

Table 5: Per-site mIOU segmentation performance for *HRNetV2 w/ Real Supervision* trained on real data. Last column is averaged across all 14 sites.

Site	Barge #1	E.B.A	Flint	Grecian	Haltiner*	H.F.*	Lucy	Monrovia	Reef*	Rend*	Thew*	Viator	Wilson	Montana	IOU_{ship} ↑
HRNetV2 [□]	0.16	0.54	0.49	0.39	0.06	0.02	0.54	0.51	0.11	0.08	0.39	0.52	0.82	0.67	0.38

We conclude that our network trained only on a large dataset of synthetic images outperformed a state-of-the-art segmentation network trained on a dataset of real, labeled shipwreck images.

3.5 UDA with Real Shipwrecks

To ensure a fair comparison with other methods in Table (3), we did not provide HRDA with real shipwrecks. However, a small quantity of unlabeled side scan sonar images of shipwrecks exist online. We collected an unlabeled dataset of 78 shipwrecks from Google Images and retrained HRDA. The model was evaluated on the same test set from TBNMS. This model achieved IOU_{ship} of 0.15 and F1 Score of 0.24. Since the shipwrecks in the online dataset are very different from those seen in the test set, it is possible that exposure to these real shipwrecks decreased performance. In general, it is very difficult to find side scan sonar images of targets online, making this method of UDA not viable for rare objects underwater.

Table 6: UDA Performance with Access to Real Shipwrecks

Method	mIOU	F1 Score
HRDA w/ Real Ships	0.15	0.24

3.6 Loss Ablation

We conducted an ablation study to learn the importance of each loss. Experimentally, we achieved best performance with a scaling factor 1 for each loss. Results are shown in Table (7).

3.7 Model Parameter Comparison

Network details are shown in Table (8). All times were measured on a single NVIDIA GTX 3090 GPU.

Table 7: Ablation Studies for Losses

Method	IOU$_{ship}$
L_{seg}	0.30
$L_{seg} + L_p$	0.35
$L_{seg} + (L_{mag} + L_{ang})$	0.33
$L_{seg} + (L_{mag} + L_{ang}) + L_p$	0.42

Table 8: # Parameters, # trainable parameters, and inference times.

Method	# Params	Inference Time (s)
PatchCore	44.7M	5e-2
HRDA	84.7M	1.25
PODA	39M	1.8e-2
Yang et. al	2.7M	7e-3
HRNetv2	70.4M	4.9e-2
Inspyrenet	90.7M	1.1e-1
STARS	84M	2.7e-2

3.8 Limitations

Training only on simulated data then transferring to real data significantly reduces the cost and manpower needed to train machine learning models. However, sonar imagery of the real world can still be very unstructured and different from simulation regardless of efforts to perform sim-to-real transfer. We notice failure cases in distractor sites and extreme debris fields like *Reef* and *Heart Failure*. For the Reef site, there are natural reef formations that look like ships, leading to incorrect labels. The Heart Failure site has a unique debris field that poses a challenge for all networks evaluated.

3.8.1 Zero Shot Transfer vs. Few Shot Learning

The key benefits of training only on simulated data (zero-shot) are that we do not need to collect expensive sonar data for the support set and we do not need to label the support set. Given the cost/difficulty of collecting and labeling sonar data, we decided to focus on the zero-shot formulation first. Finally, if the vision task is changed from shipwreck detection to an arbitrarily rare object, we would need data of that object, which is expensive and difficult to acquire. We wish to explore few-shot learning methods in future work.

3.8.2 Simulation Data Quality

Sonar imagery quality is highly dependent on environmental conditions, viewing angle, and robot trajectory. However, a practical metric for determining the quality of a synthetic image is by training a network only on synthetic images and testing it on real images. We notice this in our naïve sim-to-real baseline (HRNetv2), which achieves reasonable performance trained only on simulated data. Since there is no standard open source side scan image simulation framework, we are unable to compare our synthetic data with baselines. However, future work will consider the impact that the synthetic data fidelity has on the network performance.

3.8.3 Transferring between Sonar Sensors

Although the general image formation model of side scan sonar is similar across manufacturers, individual sensor parameters like beam angle, frequency, filtering algorithms, and data format are very different. There is no agreed upon industry standard. In this work, we showed that we can transfer more effectively from simulation to real sonar data collected

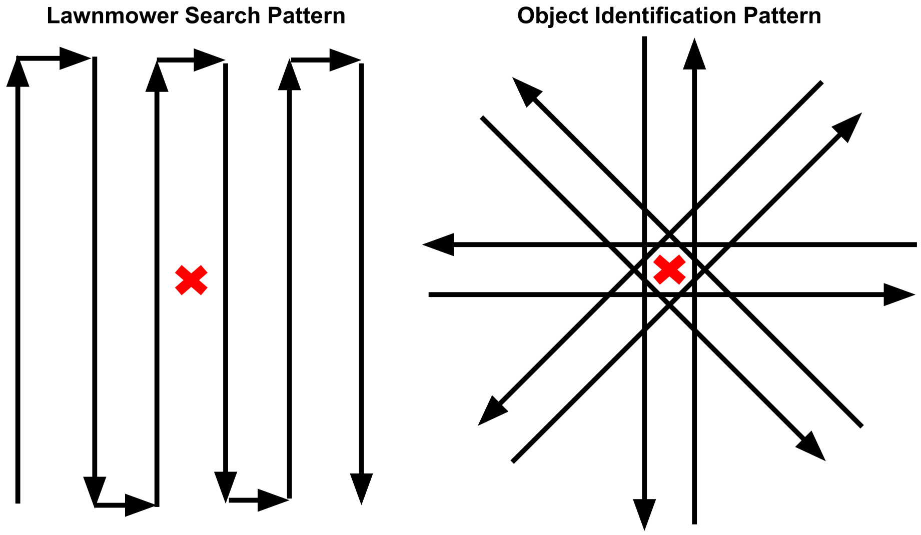

Figure 2: Lawnmower and Object Identification Patterns used for capturing multiple views of a shipwreck (shown as red X).

with an EdgeTech 2205 side scan sonar. Since acquiring another sonar sensor is very expensive, we will explore the generalizability in future work.

4 TBNMS Dataset Information

Exemplar shipwreck sites from our diverse and challenging TBNMS dataset can be found at our project webpage: `https://umfieldrobotics.github.io/STARS.github.io/`. Our dataset consists of 220 scans of 14 distinct sites of varying destruction levels. Labels were generated by an expert marine archaeologist from Thunder Bay National Marine Sanctuary. Note that some sites are mere debris fields while others are better preserved. Given the high resolution nature of side scan sonar imagery, we are able to zoom in closely and create labels for images at their native resolution of 1728 × 1728.

Shipwrecks were captured using an EdgeTech 2205 side scan sonar mounted onto an Iver3 autonomous underwater vehicle shown in Figure (3). Side scan sonar imaging is view-angle dependent, so multiple views of the same site were captured using lawnmower patterns and object identification patterns (OID) shown in Figure (2).

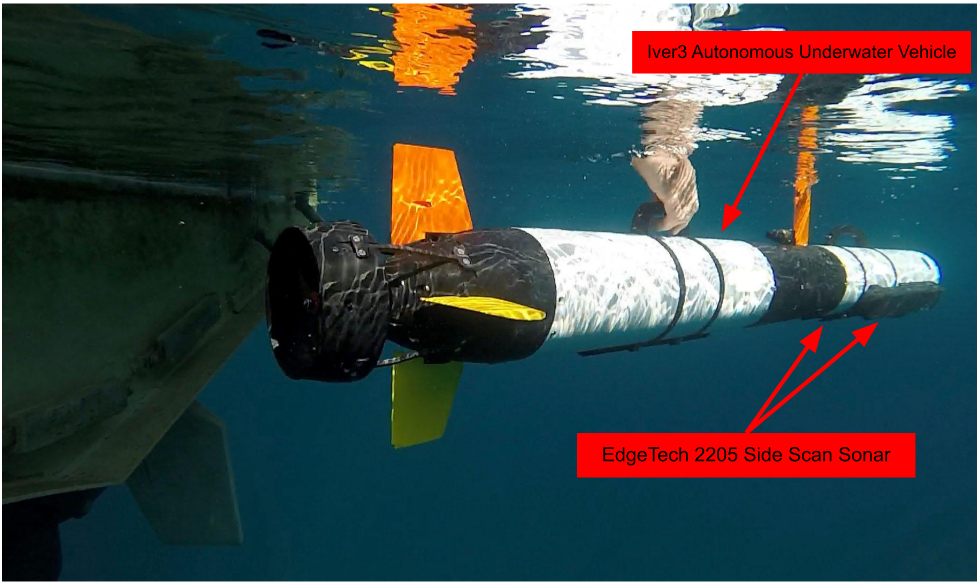
Iver3 Autonomous Underwater Vehicle
EdgeTech 2205 Side Scan Sonar

Figure 3: Iver3 autonomous underwater vehicle (AUV) being deployed to collect side scan sonar data. The AUV follows search trajectories like those shown in Figure (2). Some details removed to preserve anonymity.

References

[1] Antoni Burguera and Francisco Bonin-Font. On-line multi-class segmentation of side-scan sonar imagery using an autonomous underwater vehicle. *Journal of Marine Science and Engineering*, 8(8):557, 2020.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[3] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624, 2020.

[4] Mohammad Fahes, Tuan-Hung Vu, Andrei Bursuc, Patrick Pérez, and Raoul de Charette. Pøda: Prompt-driven zero-shot domain adaptation. In *ICCV*, 2023.

[5] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[6] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, page 372–391, Berlin, Heidelberg, 2022.

[7] Taehun Kim, Kunhee Kim, Joonyeong Lee, Dongmin Cha, Jiho Lee, and Daijin Kim. Revisiting image pyramid structure for high resolution salient object detection. In *Proceedings of the Asian Conference on Computer Vision*, pages 108–124, 2022.

[8] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2806–2814, 2021.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.

[10] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14318–14328, June 2022.

[11] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5686–5696, 2019.

[12] Dianyu Yang, Chensheng Cheng, Can Wang, Guang Pan, and Feihu Zhang. Side-scan sonar image segmentation based on multi-channel cnn for auv navigation. *Frontiers in Neurorobotics*, 16:928206, 2022.

[13] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, page 173–190, Berlin, Heidelberg, 2020.