A semantic volumetric segmentation and boulder detection framework for high-resolution sub-bottom imagery using a 3-D U-Net convolutional neural network

1st Shannon-Morgan Steele
Kraken Robotics
Dartmouth, Nova Scotia, CA
https://orcid.org/0000-0001-5462-9027
ssteele@krakenrobotics.com

2nd Maria Kotsi
Kraken Robotics
St. John's, Newfoundland, CA
https://orcid.org/0000-0003-3153-346X

3rdRyan Laidley
Kraken Robotics
St. John's, Newfoundland, CA

4th Michael Manning
Kraken Robotics
St. John's, Newfoundland, CA

5th Jacques Y. Guigne
Kraken Robotics
St. John's, Newfoundland, CA

6th Stephanie Abbott
Kraken Robotics
St. John's, Newfoundland, CA

***Abstract*—The Sub-Bottom Imager™ (SBI) is a sub-bottom inspection tool that offers 3-D real-time imaging. The SBI has a well-established commercial track record of accurate detection and positioning of buried boulders and potential unexploded ordinance (p-UXO), validated through directed field verifications and excavations. However, inefficiencies can arise during the target picking and interpretation phase, as a considerable amount of human and computer interaction is required. With the growing increase in data collection and acquisition time, this has become a time-consuming process. To address this, a 3-D U-Net convolutional neural network that performs volumetric image segmentation was developed. The model was trained with sub-bottom imagery from geohazard surveys collected during previous SBI campaigns and their associated binary labels. Due to the significant data imbalance, the mean intersection over union (mean IoU) was used to evaluate the accuracy of the predictions. The initial results have been encouraging, achieving a mean IoU of 70 % and 60% on the training and validation data, respectively. Analysis of the model predictions show the model is capable of distinguishing anomalies from the seabed response, noise, and other large scale artifacts. This preliminary work demonstrates our initial results, lessons learned, and future directions and developments. To our knowledge, this is the first application of 3-D U-Net to a sonar application.**

***Index Terms*—sonar, synthetic aperture, volumetric imaging, sub-bottom imaging, U-Net, 3-D U-Net**

I. Introduction

Buried anomaly (boulder and p-UXO) detection is required for a variety of applications including oil and gas infrastructure construction and maintnence, wind turbine installation, and infrastructure decommissioning. Such applications typically require high imaging resolution over large areas. High resolution at high area coverage rates can be efficiently achieved through SAS processing. SAS enables high spatial resolution by coherently combining acoustic returns as the sonar platform moves in the along-track direction to synthesize a large virtual array. The resolution obtained using SAS is independent of the sensor's range and the depth to the target.

The SBI, developed by PanGeo Subsea, is a novel system that provides high-resolution sub-bottom investigations with real-time 3-D imaging. Its novelty relies on the combination of beamforming with SAS processing, and the use of an inertial navigation system (INS) that allows for accurate source and receiver positioning and orientation. For its source, the SBI utilizes three chirp sonar projectors that sweep through frequencies of 4.5 - 12.5 kHz. For its receivers, it employs a 40-channel linear hydrophone array that has an overall length of 3 m (Figure 1). The SBI uses an INS that provides accurate positioning and orientation of the hydrophone array when each chirp transmits a pulse. Depending on the survey requirements, the system's frame can be adjusted to fit onto an ROV or mounted to a towfish. The SBI system can image, in real-time, a 5 m wide by 8 m deep section of the seabed while moving with a speed of 1.6 m/sec and flying at a height of approximately 3.5 m above seabed. The SBI velocity is limited to 1.6 m/sec to avoid spatial aliasing in the imagery. The SBI utilizes a delay and sum beamformer with fixed aperture processing in the across-track direction and synthetic aperture processing in the direction of travel. In the across-track direction, the SBI achieves a resolution of 3.3°, thus the across-track resolution degrades with depth in the seabed from 20 cm at the interface to 60 cm at 8 m depth. SAS processing in the along-track direction yields a range independent resolution of 20 cm and the depth resolution of the SBI is approximately 9 cm.

In addition to environmental and electrical noise, sub-bottom imagery features many different types of scattering phenomena (sediment layers, heterogeneous seabed, multipath, etc.) that appear similar to target anomalies (geohazards and p-UXOs) or occlude the target anomalies. Due to this, anomaly detection typically requires subject matter expertise with training in interpreting SBI data. The anomaly selection process is time consuming; it is repetitive, yet tedious, making it a high priority task to automate. Due to the challenging nature of the dataset, deep learning is best suited for the task. Through deep learning, it is expected that SBI anomaly detection will be faster (more cost effective), safer (less time spent operating offshore), and more consistent (less subjective and reduced or at least consistent biases). With these objectives in mind, we develop a volumetric deep learning architecture based on the U-Net model [1]. The U-Net architecture has been successfully extended to volumetric imagery in the medical imaging community [2], [3]; however, to the authors' knowledge this is the first application of the U-Net architecture to volumetric seabed imagery. This paper will focus on presenting techniques for maximizing the predictive power from challenging datasets that are class imbalanced, noisy, and small.

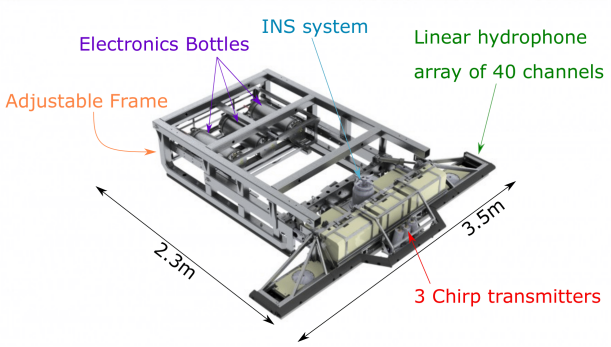

Fig. 1. Diagram of the Sub-Bottom Imager™ (SBI)

II. Methods

A. Training Data

The training and validation data consisted of 250 image volumes of the size 3 m x 10 m x 8 m in the across track, along track, and depth directions, respectively (Figure 2, top). Each imaging volume contained a random number of discrete anomalies (boulders and p-UXOs) with various sizes. The image volumes were converted to voxelized matrices for input into the 3-D U-Net model. The bottom image in Figure 2 shows a y and z (depth) axis slice through a target anomaly located in the voxelized matrix at approximately -2 m depth and centered around 5.2 and 3.6 m in the x and y axis. Training labels were generated using the python-based graphical image annotation tool *labelme* (https://github.com/wkentaro/labelme). To use *labelme*, each voxelized matrix was sliced into 2-D imagery along the depth axis. The 2-D image slices were then used for labeling regions as target (anomalies present) and background (no anomalies present). Labelling was performed by subject matter experts. The binary labelled images were then stacked back into a volumetric matrix of equal size and shape as the training image.

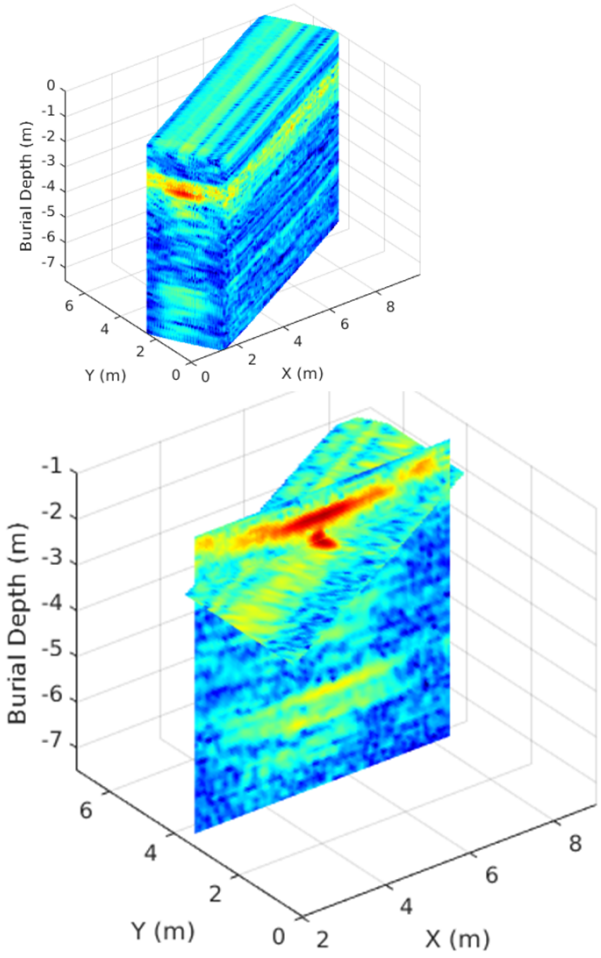

Fig. 2. Top: Rendering of a SBI seabed volumetric image used to train the 3-D U-Net model. Bottom: 2-D slices along the y and z (depth) axis to show target anomaly located inside the volumetric image above.

B. Model Architecture

We use a fully connected convolutional neural network (FCN) [4] to perform volumetric image segmentation. Here, we adapt the U-Net architecture [1] to be suitable for three dimensional acoustic imagery, we refer to this architecture as 3-D U-Net. We chose to follow the U-Net architecture because it is considered one of the top performing semantic segmentation architectures in the biomedical imaging community, with a track record of performing well with small training sets [5]. U-Net has an encoder/decoder architecture; the encoder follows the contracting path and is effectively a classification network and the decoder follows the expanding path. The encoder part of the U-Net architecture consists of convolutions and max pooling operations that capture fundamental characteristics of the input image into low-resolution feature maps. The

decoder part takes these low-resolution maps and upscales them to high-resolution feature maps by passing them through transposed convolutions and concatenation operations.

C. Training Pipeline

The image volumes are high density and thus memory intensive. To efficiently store, access, and process the image volumes on the fly we utilized TensorFlow TFRecord files (https://www.tensorflow.org/tutorials/load_data/tfrecord). The TFRecord format utilizes protocol buffers to efficiently serialize structured data. The TFRecord format can store multiple image volumes and their corresponding labels in one file. For efficient file reading, we follow the Tensorflow recommendation of storing our data in multiple TFRecord files of size 100 MB. This equates to storing 4-6 voxelized image volumes, their corresponding masks, and metadata (volume shape and sample name). The TFRecord files were written in the order that they were processed and labelled. This means each TFRecord file includes image volumes from similar imaging regions. Data shuffling is required to ensure 1) the training data is representative of the entire dataset and 2) there is no discernible order to the data as it's presented to the model. To sufficiently shuffle the data in the TFRecord files we start by shuffling the order of the TFRecord files, followed by an interleaving of the records from different files. These records are then split into 80 % for training and 20 % for validation. We can then parse 40 records from each of the training and validation datasets to be preprocessed and shuffled before being fed into the model. For rapid processing and memory efficiency we utilize prefetching to simultaneously prepare the data and train the model. The prefetching was enabled by parsing and preprocessing on the CPU while training on the GPU. The training was done on a microserver equipped with an NVIDIA GeForce RTX 2080 Ti R GPU with 11 GB of RAM and a memory speed of 14 Gbps. The preprocessing steps included:

1) Normalization of each image volume
2) Resizing each dataset to a cube, as needed by the 3-D U-Net. Here, we use a cube size of 96x96x96 pixels.
3) Application of augmentations: random gamma, gaussian noise, and random 90 degree rotations.

D. Feature Engineering

To suppress noise and other phenomena in the imagery the first feature engineering technique we tested was applying a contrast adjustment (i.e. dynamic scaling adjustment). We tested multiple different dynamic scaling options through qualitative, visual analysis; we found the best performing dynamic scaling to be a sigmoid correction [6], with a gain of 10 and a cutoff 0.85. As will be demonstrated in the results section, the model was first trained to perform binary semantic segmentation. However, we found the performance of the binary segmentation to be unsatisfactory. We suspect that one of the reasons why the model performs poorly on binary segmentation is because of the extreme class weighting we must apply to account for the dataset imbalance. Since 99.96 % of the voxels in the dataset are background, we applied a weighting of 4×10^{-4} to all the background pixels. This means the majority of the voxels are weighted to nearly 0, which is effectively equivalent to training on the target voxels only, which is a very small amount of data overall. To test and possibly overcome this class weighting issue we introduced artificial classes to the dataset. Artificial classes were generated in an unsupervised manner by splitting the background voxels into multiple different classes based on their intensity. The background class was split into ten artificial background classes to form an 11-class semantic segmentation problem, which increases the weighting for the background voxels by a full order of magnitude. Introducing artificial classes to the training data changes the segmentation problem from binary to multi-class, meaning the model inputs, activation functions, and loss type must change accordingly. In section III we will present model training results from three different versions of the model: no feature engineering applied, sigmoid correction applied, and sigmoid correction and artificial classes applied.

E. Hyper–parameter Selection

Hyper–parameter selection is an important task for maximizing model performance. We tested a number of different parameters, the best performing parameters were:

- **Regularization:** we found $l2$ regularization performed better than $l1$ regularization.
- **Dropout and Batch Normalization:** We tested a range of dropouts from 0.1 to 0.5, with and without batch normalization applied. We found the best combination was to apply batch normalization and a dropout of 0.2.
- **Activation:** Sigmoid (for binary segmentation) and Softmax (for multi-class segmentation)
- **Optimizer:** we tested a number of optimizers — Adam, NAdam, RMSprop, SGC— and found that the Adam optimizer with a learning rate of 1×10^{-4} worked the best in this problem.
- **Loss:** We utilized binary Cross-entropy (for binary segmentation) and Categorical Cross-entropy (for multi-class segmentation)

F. Model Evaluation

To evaluate the accuracy of our prediction we used the mean intersection over union (mean IoU). This is a common evaluation metric for semantic image segmentation tasks with unbalanced datasets. While some metrics, such as accuracy, are sensitive to dataset imbalances, the mean IoU is class imbalance insensitive because it computes the intersection over union for each class and then calculates the average over classes. The IoU is defined as follows:

$$IoU = \frac{TP}{TP + FP + FN}, \tag{1}$$

where TP is number of true positives, FP in the number of false positives, and FN is the number of false negatives. For the case of artifical classes, before computing the mean IoU, we converted all the background classes back to one class, which we refer to as the masked mean IoU.

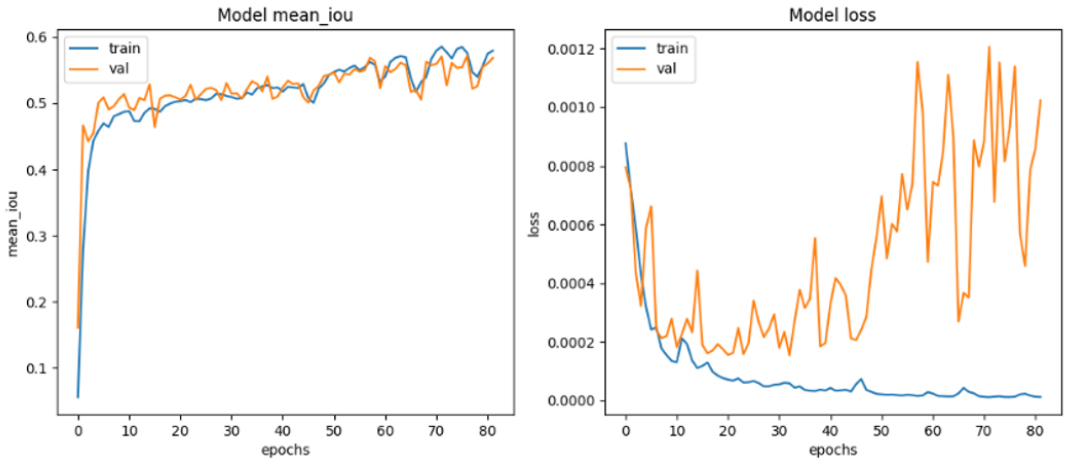

Fig. 3. Mean IoU and loss for model trained on binary masks without sigmoid correction to the imagery.

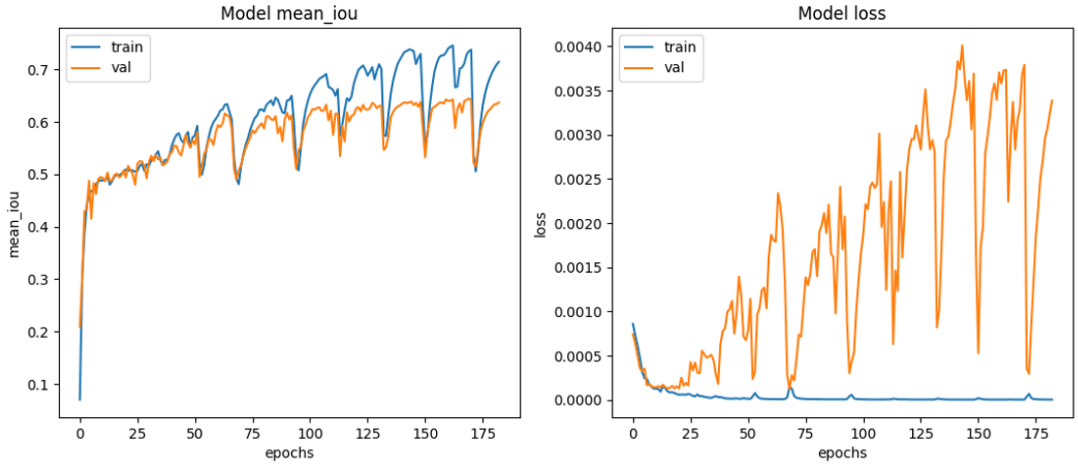

Fig. 4. Mean IoU and loss for model trained on binary masks with sigmoid correction to the imagery.

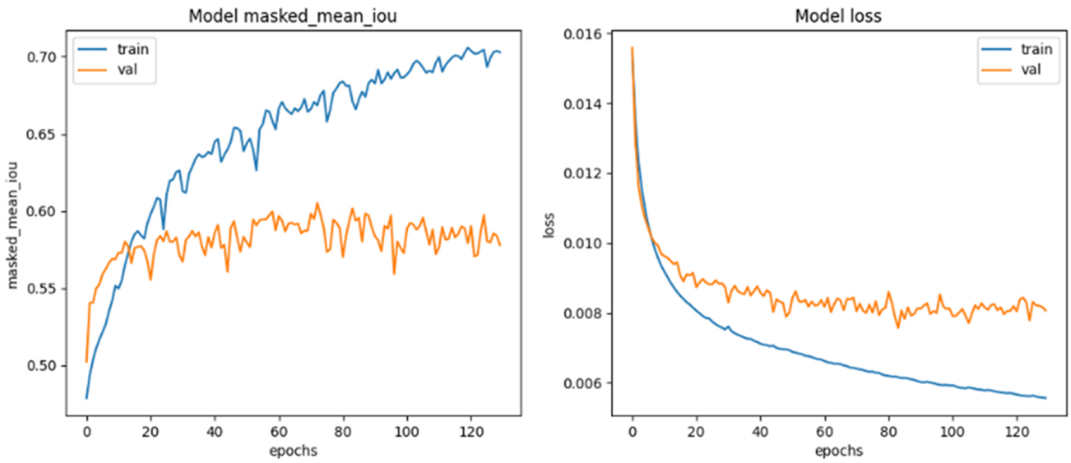

Fig. 5. Mean IoU and loss for model trained on multi-class masks using artificial classes and sigmoid corrected imagery.

III. Results and Discussion

Each of the three versions of the model were trained over 250 epochs with early stopping on the performance plateau. While the model without any feature engineering techniques applied shows some evidence of learning, it rapidly plateaus at an unsatisfactory performance level in terms of the overall achieved mean IoU (55 %) as well as the validation data loss, which begins to increase after 10 epochs, indicative of over fitting (Figure 3). The model that includes the sigmoid scaling feature engineering technique shows an improved performance in comparison to the previous model: the model reaches a much higher mean IoU of 72 % and 60 % on the training and validation data, respectively. But, the validation loss begins to increase after about 25 epochs (Figure 4), indicating the model is over fitting again. We also observe in Figure 4 that there is a distinct undulation pattern for both the training and validation mean IoUs as a function of epoch. Due to the black box nature of neural networks we cannot know for certain what is causing these undulations; however, we suspect they are indicative that the model may be "stuck" in a local minima within the cost function. It's possible that the gamma scaling contributed to this behaviour by emphasizing targets as well as some other non-target anomalies. When artificial classes are applied to the sigmoid scaled volumes, we see a significant improvement in the model behaviour. The loss function is now smooth and consistent for both the training and validation data (Figure 5); however after about 20 epochs we see the model start to overfit. In terms of the mean IoU the model reaches a maximum of 70 % and 60 % for the training and validation data respectively.

We consider the model with both sigmoid scaling and artifical classes to be our best performing model, and thus we only show results from that model. Here, we present some examples of 2-D image slices and their corresponding training and prediction masks (Figure 6). The algorithm successfully predicts anomalies when they're present and successfully identifies other large artifacts or seabed anomalies as background. However, the model struggles to distinguish target anomalies from small artifacts in the imagery (Figure 6, third row). The performance level of the model is not quite satisfactory for utilization as a commercial tool; however, we believe the initial results indicate that with further development the performance can be improved to meet industry requirements. This further development is currently ongoing and includes: an expanded training dataset, refinements to the training data, improved image processing techniques to reduce noise, stricter data and label quality analysis, methods for simulated data generation, and additional augmentation techniques.

IV. Conclusion

In this paper we have presented an adaptation of the U-Net model for semantic segmentation of 3-D (volumetric) sonar imagery. By introducing two feature engineering techniques (sigmoid correction and artificial classes) we are able to significantly improve the model loss behaviour, improving the mean IoU by 15 % and 5 % for the training and validation data respectively. The model can easily distinguish between targets and large scale anomalies; however, it often falsely identifies smaller artifacts as target anomalies. While the initial result of a declining loss function and mean IoU at 60% is encouraging, more development is required for the model to be useful operationally. Future work will focus on preventing the model from over fitting by suppressing noise in the imagery, as well as exploring additional augmentation techniques and simulation techniques.

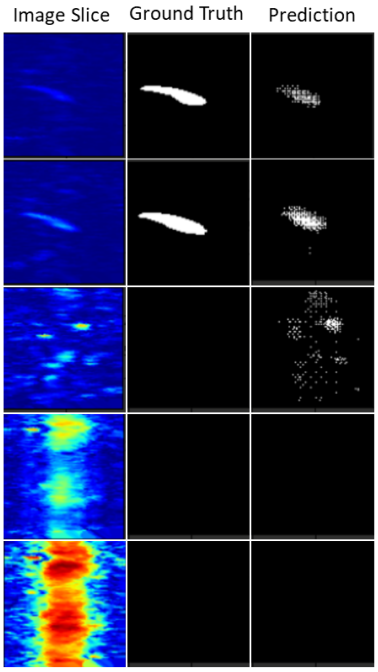

Fig. 6. Examples of model predictions; left column contains the image slices, the middle column represents the ground truth label, and the right column represents the 3-D U-Net prediction. In the ground truth and predictions white indicates target and black is background.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *International Conference on Medical image computing and computer-assisted intervention-MICCAI 2015*, 2015.

[2] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," *CoRR*, vol. abs/1606.06650, 2016.

[3] W. Zhao, D. Jiang, J. Peña Queralta, and T. Westerlund, "Mss u-net: 3d segmentation of kidneys and tumors from ct images with a multi-scale supervised u-net," *Informatics in Medicine Unlocked*, vol. 19, p. 100357, 2020.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," IEEE Conference on Computer Vision Pattern Recognition, 2014.

[5] J. Kugelman, J. Allman, S. A. Read, S. J. Vincent, J. Tong, M. Kalloniatis, F. K. Chen, M. J. Collins, and D. Alonso-Caneiro, "A comparison of deep learning u-net architectures for posterior segment oct retinal layer segmentation," *A comparison of deep learning U-Net architectures for posterior segment OCT retinal layer segmentation*, vol. 12, 2022.

[6] G. J. Braun and M. D. Fairchild, "Image lightness rescaling using sigmoidal contrast enhancement functions," in *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts IV* (G. B. Beretta and R. Eschbach, eds.), vol. 3648, pp. 96 – 107, International Society for Optics and Photonics, SPIE, 1998.