Deep Learning in Shallow Water: CNN-based 3D-FLS Target Recognition

Heath Henley
Software Engineer
FarSounder, Inc.
Warwick, RI
heath.henley@farsounder.com

Austin Berard
Software Engineer Intern
FarSounder, Inc.
Warwick, RI
austin.berard@farsounder.com

Evan Lapisky
Software Engineering Manager
FarSounder, Inc.
Warwick, RI
evan.lapisky@farsounder.com

Matthew Zimmerman
VP of Engineering
FarSounder, Inc.
Warwick, RI
matthew.zimmerman@farsounder.com

***Abstract*—Automatic target detection is important for real time navigation applications of 3D forward looking sonar (3D-FLS) systems. A 3D-FLS sensor generates a large volumetric point cloud of data that updates on the order of seconds, so that manually interpretting the data is not feasible for a vessel operator. FarSounder has developed an algorithm for detecting two classes of targets (seafloor and in-water-targets) based on traditional image and signal processing techniques. In this work, two modified versions of CNN architectures previously developed for volumetric data (3D U-Net and 3D V-Net) are evaluated for their ability to replace the current detection algorithm. The current detection alorithm was used to generate the set of training data and validation data for training and evaluation. Additional models were developed based on the 3D U-Net and V-Net models to operate on 2D cross sections of the data volumetric data as input instead of the volumetric input. All of the volumetric models acheived higher validation and training accuracy than the 2D versions, and the 3D U-Net replicated the traditional algorithm most closely. Finally, an automated procedure for improving training data using NOAA bathymetry data is described.**

I. Introduction

Automatic target recognition is very important for a number of sonar based systems, particularly for 3D forward looking sonar (3D-FLS) system when used for navigation in real-time. The navigational 3D-FLS system used in this study insonifies the volume ahead of the vessel (up to 1000m forward at 60 deg to port and starboard, or 500m forward at 90 deg to port and starboard) and collects the backscatter levels using a bow mounted transducer array. The resulting data is a 3D point cloud of target strength values in front of the vessel, refreshing after each ping (1.6 seconds).

Figure 1 depicts a visualization of the resulting array processed data in 3D, with color mapped to target strength in dB. Clearly, inspecting the raw sonar data for the location of the seafloor and any navigational hazards would require a great deal of operator training and attention. Given that the entire dataset refreshes every 1.6 seconds, manually monitoring this data and recognizing targets and making navigation decisions is a demanding task. At FarSounder Inc., this motivated the implementation and development of automatic target recognition algorithms to increase ease of use and comprehension of navigational 3D-FLS data. Note that automatic target recognition algorithms are even more essential for autonomous applications of navigational 3D-FLS.

A variety of target detection or target recognition methods have been developed for sonar applications. Some traditional methods are based on determining peaks in sonar returns using either fixed or adaptive thresholding. Other methods apply image processing techniques such as edge detection in combination with thresholding in order to algorithmically detect targets. Wang et al. (2017) [1] presented a recent review of target detection and feature extraction methods in underwater sonar in general. Greethalakshmi et al. (2011) [2] presented a review of target detection in side-scan sonar images using image processing techniques, specific to detecting mine like objects. Recently, especially in the last few years, convolutional neural networks (CNNs) have been applied to a number of image segmentation and object detection problems with very promising results. Naturally, CNN based frameworks have also been applied to sonar data as well. For example, Dzieciuch et al. (2016) [3] developed a simple CNN framework for detection of mine-like objects in sonar images that obtained an accuracy of 99% in the conditions of their study. Further Kim et al. (2016) [4] presented a CNN based algorithm for detection of a small ROV in traditional forward looking sonar images. Valdenegro-Toro (2017) [5] investigated the accuracy of three CNN models for object detection in traditional forward looking sonar images as a function of object and training set size, and investigated how well transfer learning using developed CNN models performed on forward looking sonar data. Finally, Livne et al. (2018) [6] presented a review of the current challenges associated with using CNNs for object detection in traditional forward looking sonar data. They concluded that due to the typically lower signal-to-noise ratio in forward looking sonar data, and the lack of publicly available labelled data, much more work

978-1-5386-4814-8/18/$31.00 ©2018 IEEE

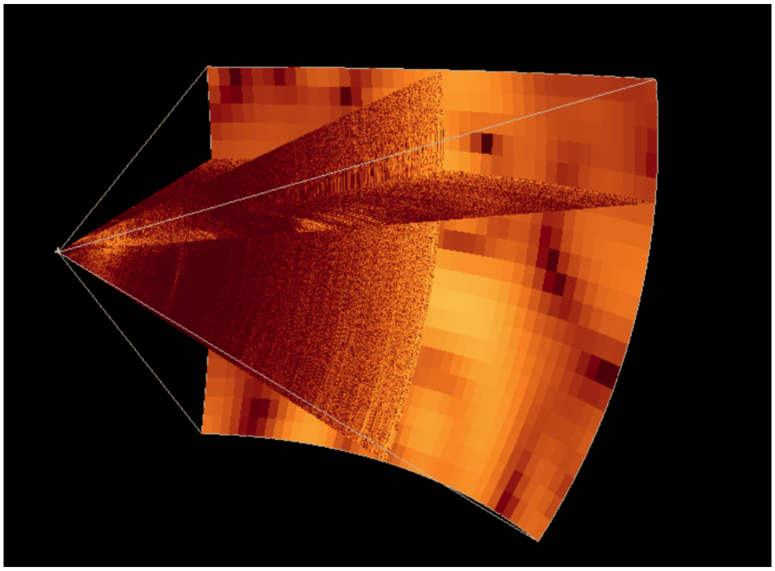

Fig. 1. 3D projection of target strength values from a single ping using a 3D-FLS. Only one single vertical, horizontal, and radial plane are displayed for clarity.

is needed to be done and that deeper networks (compared to the model they investigated) should be considered. Thus far, all of the forward looking sonar images considered in the literature (to the best of the authors knowledge) has considered short range, high resolution, 2D imaging sonars. The sensor considered in this work is a FarSounder 3D-FLS, and as introduced above (see Figure 1) it is capable of generating a 3D point cloud of 'target strength' or backscatter strength values. While 3D-FLS data has not been yet been considered, a number of researchers have applied CNN based frameworks to volumetric point clouds of data, particularly for applications related to autonomous driving and medical imaging. Maturana and Scherer (2015) [7] developed VoxNet, and applied it to publically available benchmarks using 3D LIDAR, RGBD, and CAD data. Zhou and Tuzel (2017) presented VoxelNet [8], a framework for 3D object detection, and successfully applied it to LIDAR data from the KITTI dataset. V-Net was developed by Milletari et al. (2016) [9] and applied successfully to 3D magnetic resonance images (MRI), while a 3D version of U-Net [10] was developed by Cicek et al. (2016) [11] applied to 3D confocal microscope data. Many other contributions exist and this review is not intended to be exhaustive, however, a CNN framework for object detection in 3D-FLS has not yet been presented in the literature. Therefore, the main objectives of this work are to (1) evaluate the performance of a few distinct CNN based models for target detection on 3D-FLS data compared to detection methods based on traditional image processing, and (2) investigate ways to generate and improve the training dataset to limit any requirement of manual labelling.

II. Procedure

A. Processing Overview

The 3D point cloud of backscatter results obtained using a 3D-FLS (as depicted in Figure 1) needs to processed and presented in a way that is easy to understand for end users. FarSounder's current processing chain uses traditional signal and image processing techniques to make 'detections' in the data and obtain information about the location of the seafloor and any navigational hazards ahead of the vessel. A high level outline of the processing chain is presented in Figure 2.

Due to the recent success of CNNs for image segmentation and object detection operations, it is likely that a CNN based model could entirely replace our current 'detection' algorithm (eg. red dashed box in Figure 2) and, with the proper training data, perform significantly better than the current algorithm. The network architecture is important to the efficacy of any CNN based model for a given application, and many novel architectures have been published recently for processing some other types of 3D data. In searching for the architecture that works best for 3D-FLS, a baseline dataset, for which each architecture can be compared is extremely useful. Given that nature of 3D-FLS data (a dense cloud of backscatter information with comparatively low SNR) as depicted in Figure 1, the suitability of a CNN model for this problem must first be investigated. That is, is it even possible to replace the current target detection algorithms with a CNN based model? The first step in this work is to determine whether a CNN based model can address this problem. This is accomplished by developing a 'training baseline' from a few recently developed CNN models, and comparing the results to the standard algorithm. Next the accuracy of each CNN model is compared using the same training and validation data. Finally, the top performing architecture is trained on improved

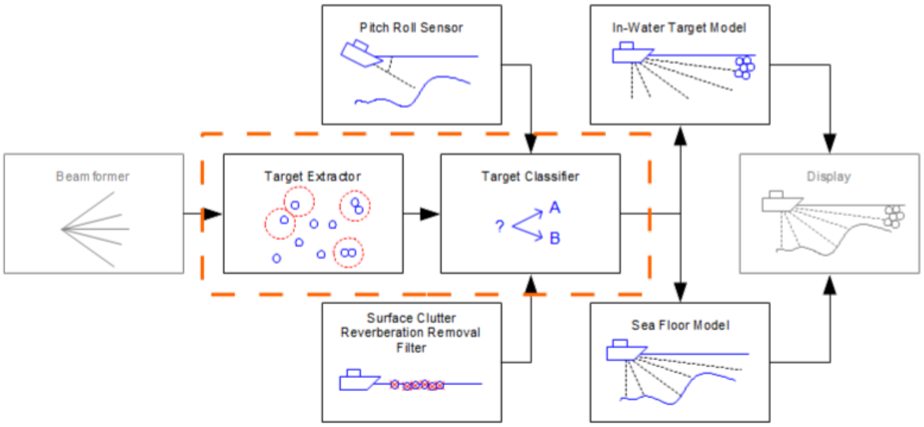

Fig. 2. Current target detection algorithm flowchart.

training data, and the results are are compared to the standard algorithm and presented.

B. Model Input

The current input to the model is a 3-dimensional array of backscatter strengths, and while the exact dimensions are proprietary, the total number of voxels is on the order of millions (see Figure 1). The task of both the traditional detection algorithm and the CNN-based model is to generate a class label for each point in this input volume. The current possible classes are either (1) background, (2) seafloor, or (3) in-water target. This produces an output with the same shape as the input in the first three dimensions but an additional dimenension of length three appended to the end. The three element vector in the last dimension represents the probability of a voxel being background, seafloor or in-water target.

C. Training Baseline

Given that a procedure (Figure 2) for turning sensor level data (Figure 1) into a human readable map of detections (seafloor and ‘in-water target’) is already in place, it was straightforward to develop a baseline for the evaluation of any potential network architectures, loss function, optimizer, and/or hyperparameter combinations. The existing processing chain was applied to limited set of data to automatically generate a labelled training set. Training a network on this data set is essentially an attempt to directly replace the current processing with the CNN model, which should be a great first indicator of the fitness of a given model for this application.

D. Architectures Considered

The data obtained from the 3D-FLS is inherently a 3D point cloud of backscatter strengths ahead of the vessel, so an intuitive approach is apply a CNN model to the data that implements 3D convolutional layers. Two architectures presented in the literature for processing volumetric data are the 3D-UNet [11] and V-Net [9] architectures. Figures 3 and 4 illustrate the original published architecture of 3D-UNet and V-Net, respectively.

In this work, slightly modified versions of both the 3D U-net and V-Net architectures are were tested. Due to the size of the input data generated by the 3D-FLS sensor, the number of filters used in the volumetric convolutions was reduced in both the U-net and V-Net models until the model could be trained successfully (with running out of memory resources) on an 8 GB Nvidia GTX 1070 GPU. Of course, better results might be observed running on a system with more available GPU memory. However, considering that the end goal of this investigation is to deploy the network for real-time target detection on computers currently specified with Nvidia GTX 1050 (with 4 GB of GPU memory). Based on our observations, the production machines using the model for predictions use about half of of the memory used in training, most likely due to storage of gradients for backpropagation, so for the purposes of this work, though it may produce a more robust model, is not beneficial.

In addition, 2D versions of these models were created by switching the 3D convolutions with 2D convolutions and tested by inputting 2D slices of the 3D input data into the model. For the same size GPU memory footprint, the 2D version of the model allow a greater number of filters to be considered at each layer, while neglecting information about the adjoining slices through the data.

E. Manually Editing Training Data

Manually labelling all of the seafloor and in-water targets would not be efficient, as indicated above. However, the current detection algorithm produces a number of detections that are not necessarily useful for the user to see. A major example of this is the wake of recently passed vessel, or directional noise from another passing vessel. To avoid training the CNN model to continue to detect these targets, as the tra-

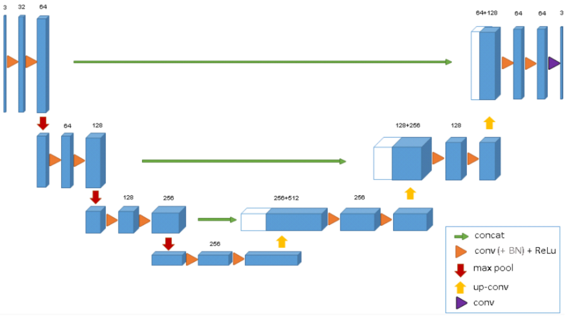

Fig. 3. 3D U-net architecture presented by Cicek et al. (2016) (reproduced). The same architecture was used in this work. The number of features layers was tuned as indicated in the text to reduce the GPU memory footprint.

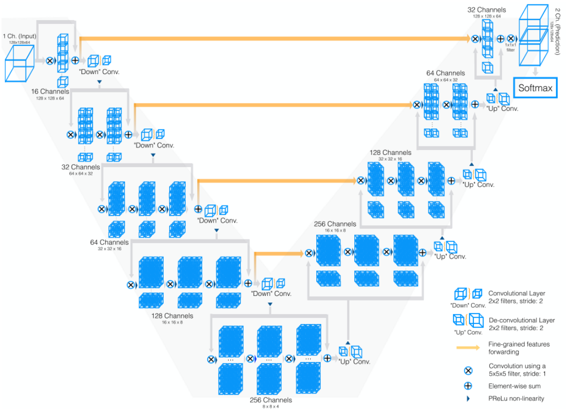

Fig. 4. 3D V-Net architecture presented by Milletari et al. (2016) (reproduced). The same architecture was used in this work. The number of features layers was tuned as indicated in the text to reduce the GPU memory footprint.

ditional algorithm does, a tool was developed to clean the data manually. So that in the future, each ping can be inspected, and any detection caused by vessel wakes or interference from engine noise will be removed from the training data.

F. Automated Training Data Improvement

One approach to improving the training data was to use survey data collected by NOAA to supplement detections of the seafloor generated by the traditional algorithm. The NOAA survey data is collected using a higher resolution sensor (MBES) and cleaned in post processing by a hydrographer, so it is likely a reasonably good estimate of the ground truth depth in the area, assuming that the area was surveyed somewhat recently. Surveys within the training data were downloaded and added to a database containing the location and depth for the survey points. For each ping the database was queried for points within the field of view of the sensor. The three subplots in Figure 5 illustrate the depth as a function of position in an area with interesting bathymetry. They represent the field of view (FOV) of a single ping of the 3D-FLS, thus this fit procedure is applied over many pings.

A Radial Basis Function (RBF) interpolator was fit using this set of depths at given latitude and longitude within the sensors field of view. For each point in the 3-D point cloud, the latitude and longitude was used to compute the depth, as predicted using the RBF interpolator. If the calculated depth fell within an arbitrarily chosen tolerance of the depth of the point in the 3D cloud, the point was labelled as a bottom. The new array of labels were saved and used to train the CNN.

III. Results

The results of our investigation into whether a CNN based model is suitable for this problem, and which architecture and parameters to use, are described in this section. It is important to note that given that there are large number of possible network architectures and combinations of hyperparameters, it is unlikely that the optimum network for this problem has been found here. However, the results presented below illustrate the performance of the models that have been adapted to this problem so far. The results should not be interpreted to suggest that any particular model investigated is generally better than the rest.

A. Baseline Training data

A dataset of 1450 3D-FLS pings was recorded for use as the baseline training set. The pings included in the dataset come from a variety of different boat trips on different days in different locations. The dataset was further split into 1187 pings for training and 263 pings for validation. Each model

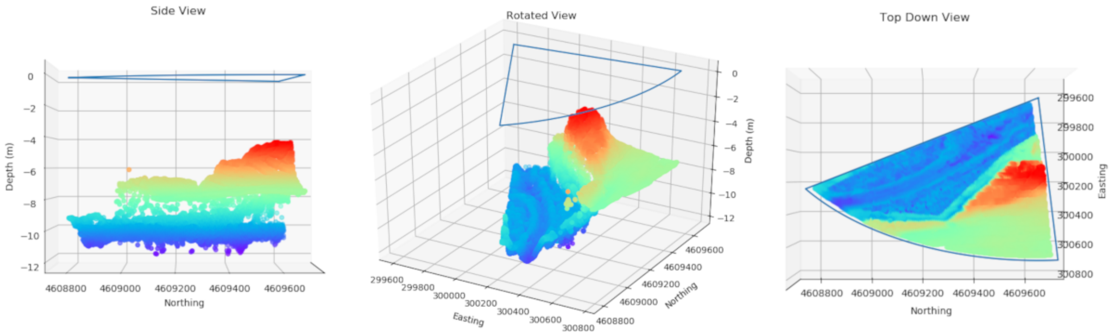

Fig. 5. NOAA survey data contained within the field of view of a single ping.

tested was implemented in python using Keras and Tensorflow. They were trained on a computer with a Nvidia GTX 1070 GPU with 8GB of dedicated memory. A batch size of 1 was used for all of the 3D versions of the CNN models, as it was required to fit the data onto the GPU. Each epoch took approximately 5 hours to complete, and the models were run for 10 epochs each. The 10 epoch limit was chosen empirically so that the loss and accuracy were not changing much from one epoch to the next. The 2D versions of the models were run on a batch size of 10 for roughly 10 epochs or until the loss became stationary, and the runtime was 1.5 hours per epoch on the same system. For both the 2D versions of UNet and VNet, one version of the model with all volumetric convolutions switched with 2D convolutions, but with the number of features kept constant. A second 2D version of each model was created with a greater number of trainable parameters by increasing the number of filters used in the convolutional layers.

The Adam [12] optimizer was used to train all models, and categorical cross entropy was chosen as the loss in all cases. Due to an imbalance in the number of background points in the 3D-FLS data compared to the number of points representing a detection, two different weighting methods were tested. The first was a sample weighting approach which used the frequency of a class within a ping to weight each sample, and in the second the weight was calculated using the frequency of a given class over the entire dataset. The later method produced models with higher training and validation accuracy. The results of running all of the models considered in this work are tabulated in Table I, where CW: Class weighted loss function, SW: sample weighted loss function, the categorical cross entropy loss function was used for all cases, and the Adam optimization algorithm with default Keras parameters was used to train the parameters. The "*" represents a model with increased features in the convolutional layers to increase the number of trainable parameters.

In regards to the quantitative metrics of accuracy over the training and validation, the 3D Vnet model with class weighted loss performed the best. However, in both the training and validation data, despite the class weighting, this model converged to predict very little of the seafloor class (the most underrepresented class). In comparison, the 3D U-net model achieved similar accuracy, but performed qualitatively better, in the sense that it predicts classes with frequencies closer to those in the training data. For this reason, the 3D U-net model using the class weighting approach was chosen as best candidate for further development out of the models investigated.

TABLE I
COMPARISON OF TRAINING RESULTS USING BASELINE TRAINING DATASET

Model	Epochs	Parameters	Training Acc.	Valid. Acc.
2D UNet*	10	385943	0.8399	0.8532
2D UNet (CW	10	41697	0.7989	0.8361
3D UNet (CW)	10	123957	0.9480	0.9459
3D UNet (SW)	10	123957	0.9456	0.8932
2D VNet* (CW)	10	216581	0.8216	0.8504
2D VNet (CW)	10	88647	0.7710	0.9111
3D VNet (CW)	10	255507	0.9507	0.9507
3D VNet (SW)	10	255507	0.9366	0.9421

An example of the qualitative performance of the model is given in Figures 6 and 7. The volumetric input data is shown on the left, with the traditional results and CNN based results in the top and bottom right respectively. The ping used to generate Figures 6 and 7 was not included in the training data set for the CNN models. In both Figures 6 and 7, the CNN models are able to detect the in-water target features that are detected using the traditional processing corresponding to the edges of a pier. However, there are are clearly some additional detections that do not exist in the training data. The CNN model used to generate Figure 7 is the same 3D U-Net model with weighted loss, however it was trained using the AdaDelta [13] optimizer, and without normalizing the input per ping. This model seems to have less 'false positive' detections. The seafloor detection generated by both CNN models in 6 and 7 agrees well with the standard algorithm at short range. However both CNN models detect additional deep seafloor

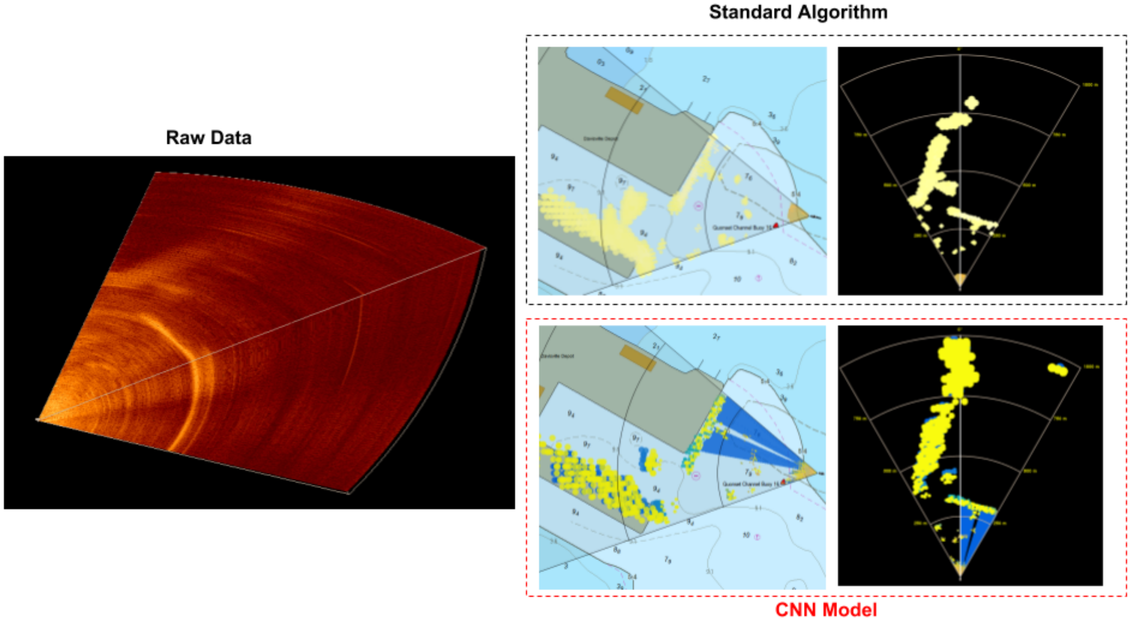

Fig. 6. Comparison of the qualitative output of the 3D U-Net model and traditional detection model for a given input.

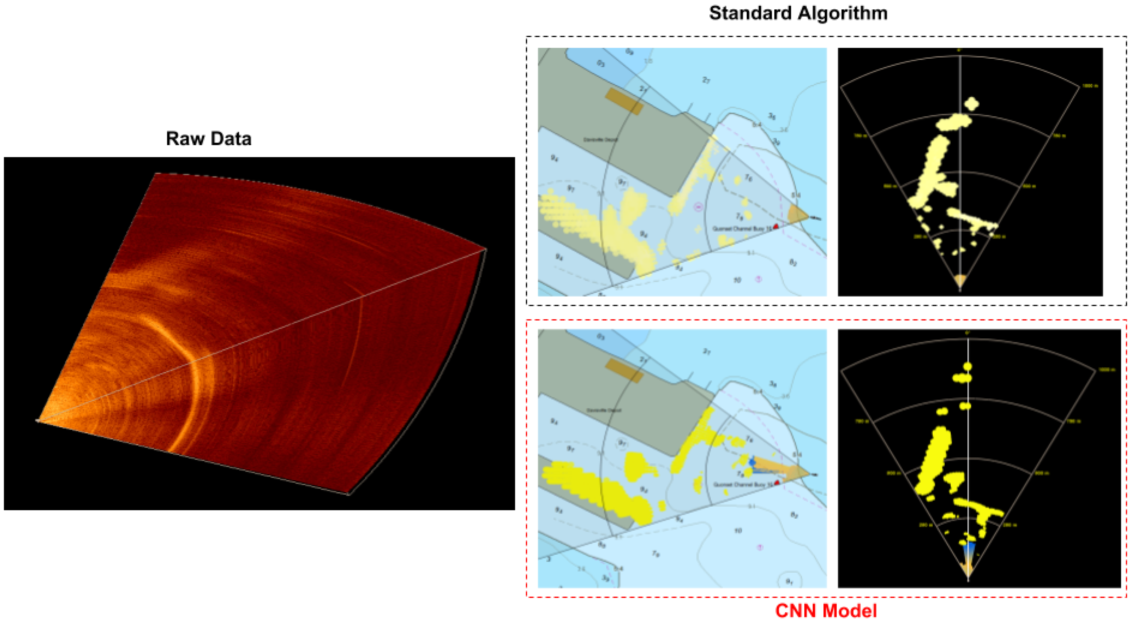

Fig. 7. Comparison of the qualitative output of the 3D U-Net model and traditional detection model for a given input. This model was trained using the AdaDelta optimizer for 5 epochs instead of Adam, and the input was not normalized per ping.

targets at longer range. The long range detections are not present in all pings, and the seafloor agrees well with the traditonal algorithm in some of the other pings investigated.

Based on the results generated by the CNN models investigated for this work, it seems that the traditional detection algorithm could certainly be replaced by a CNN model, though the CNN models need to be further refined to reduce superfluous detections, especially of the seafloor. Further, while the CNN model reproduced all of the features detected by the traditional detection algorithm, it of course replicated all of the examples of detections made by the traditional model which are not desireable (eg. engine noise from passing vessel, wakes, etc.).

A subset of the training data was improved using the automated process previously described and used to continue the training of the 3D U-Net model. However, using the current best model with the updated training resulted in poor performance and appeared to be underfitting the data. Perhaps fitting the added complexity of the longer range seafloor labels created using the NOAA survey data will require a model with a greater number of trainable parameters or the additional layers. Further, additional features could be added to the input of the CNN model in addition to only including the backscatter strength. For example, including the cartesian position of each element in the point cloud along with the backscatter strength could allow the network to handle the differences between the Cartesian and spherical coordinate representation of the data.

IV. Conclusions

In this work, a handful of CNN models were developed based on published architectures for volumetric data. A baseline set of training data was automatically generated using a detection algorithm based on traditional image and signal processing techniques, and used to train and evaluate each model. The results of training on the baseline data suggest that the current detection algorithm could be replaced by a CNN model. However, application of the CNN results in some 'false positives' that are not detected in the current processing, especially in the detection of the seafloor. In addition, if only trained on the baseline, the CNN model will of course never surpass the performance of the current detection algorithm. In order to improve the training data for the model, NOAA bathymetry data was used to improve and extend the seafloor detection in a subset of the data in the baseline training set. The added range and number seafloor points in the training data was not easily fit by the current CNN model, suggesting that additional input features, parameters or layers may be needed. Finally, a tool for manually editing the label data was developed for further improvement of the training data. The next steps in the project include generating results after training on improved data, introducing more input features to the model, and increasing the number of layers and trainable parameters to improve performance when using the training data that has been extended using NOAA bathymetric data.

References

[1] Z. Wang, W. Feng, and X. Wang, "Review and prospect of Underwater Target Feature Extraction Based on Active Sonar," *Advances in Engineering Research*, vol. 128, no. Icmse, pp. 34–37, 2017.

[2] S. Greethalakshmi., P. Subashini., and P. Geetha., "A study on detecting and classifying underwater mine like objects using image processing techniques," *International Journal on Computer Science and Engineering*, vol. 3, no. 10, pp. 3359–3366, 2011.

[3] I. Dzieciuch, D. Gehardt, C. Barngrover, and K. Parikh, "Non-linear Convolutional Neural Network for Automatic Detection of Mine-Like Objects in Sonar Imagery," in *Proceedings of the 4th International Conference on Applications in Nonlinear Dynamics (ICAND 2016)*, 2016, pp. 309–314.

[4] J. Kim, H. Cho, J. Pyo, B. Kim, and S.-c. Yu, "Convolutional Neural Network-based Real-time ROV Detection Using Forward-looking Sonar Image," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 396–400.

[5] M. Valdenegro-Toro, "Best Practices in Convolutional Networks for Forward-Looking Sonar Image Recognition," 2017. [Online]. Available: http://arxiv.org/abs/1709.02601

[6] A. Livne, A. Baruch, and H. Guterman, "Thoughts on object detection using convolutional neural networks for forward-looking sonar," vol. 4, no. 3, pp. 182–184, 2018.

[7] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition," *Iros*, pp. 922–928, 2015.

[8] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," 2017. [Online]. Available: http://arxiv.org/abs/1711.06396

[9] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," pp. 1–11, 2016. [Online]. Available: http://arxiv.org/abs/1606.04797

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," pp. 1–8, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[11] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-net: Learning dense volumetric segmentation from sparse annotation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9901 LNCS, pp. 424–432, 2016.

[12] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," pp. 1–15, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[13] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," 2012. [Online]. Available: http://arxiv.org/abs/1212.5701