Machine Learning Techniques for AUV Side Scan Sonar Data Feature Extraction as Applied to Intelligent Search for Underwater Archaeological Sites

Nandeeka Nayak, Makoto Nara, Timmy Gambin, Zoë Wood and Christopher Clark

August 25, 2019

Machine Learning Techniques for AUV Side Scan Sonar Data Feature Extraction as Applied to Intelligent Search for Underwater Archaeological Sites

Nandeeka Nayak, Makoto Nara, Timmy Gambin, Zoë Wood and Christopher M. Clark

Abstract This paper presents a system for the intelligent search of shipwrecks using Autonomous Underwater Vehicles (AUVs). It introduces a machine learning approach to the automatic identification of potential archaeological sites from AUV-obtained side scan sonar (SSS) data. The site identification pipeline consists of a series of stages that set up for, run, and process the output of a convolutional neural network (CNN). To alleviate the issue of training data scarcity, i.e. the lack of SSS data that includes shipwrecks, and improve the performance at testing time, a data augmentation stage is included in the pipeline. In addition, edge detection and other traditional image processing feature extraction methods are used in parallel with the CNN to improve algorithmic performance. Experiments from two multi-deployment shipwreck search expeditions involving actual AUV deployments along the coast of Malta for data collection and processing demonstrate the pipeline's usefulness. Results from these two field expeditions yielded a precision/recall of 29.34%/97.22% and 32.95%/80.39% respectively. Despite the poor precision, the pipeline filters out 99.79% of the area in data set A and 99.31% of the area in data set B.

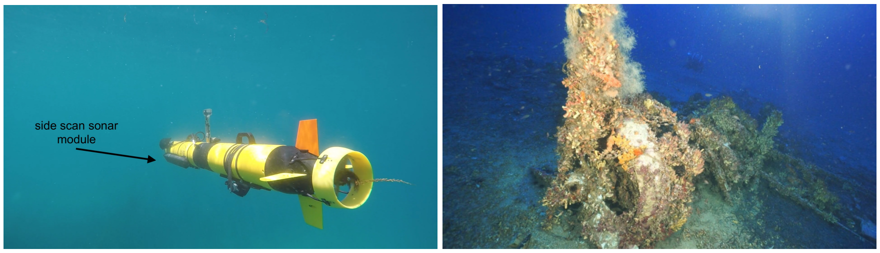
1 Introduction

Locating wrecks and sites of interest to marine archaeologists is challenging and time-consuming, since fine detail and extensive experience is required to pick out wrecks from seafloor features. Archaeologists use indicators like corners, edges,

Nandeeka Nayak, Makoto Nara, and Christopher M. Clark
Harvey Mudd College, Claremont, CA, USA e-mail: {nnayak,mnara,clark}@hmc.edu

Timmy Gambin
University of Malta, Msida MSD 2080, Malta e-mail: timmy.gambin@um.edu.mt

Zoë Wood
California Polytechnic State University, San Luis Obispo, CA, USA e-mail: zwood@csc.calpoly.edu



(a)

(b)

Fig. 1: (a) Iver3 AUV with SSS at the front and a GoPro HERO4 attached under the rear handle. (b) Remains of the World War II-era Fairey Swordfish discovered off the coast of Malta.

and the scuffing of the surrounding area to differentiate areas of interest from sand, rocks, and other natural terrain features. However, due to the high volume of data, identification of possible sites is time-intensive and inexact. Different experts will flag different objects as wrecks and occasionally contradict one another.

In order to obtain high resolution images of the seafloor, marine archaeologists often use side scan sonar (SSS). Often, SSS is used either on a towfish—where the SSS is mounted on a hydrodynamic block towed behind a boat—or, more recently, on autonomous underwater vehicles (AUVs). Once the SSS data is collected, it is converted to images and inspected for regions of interest. If a location seems to be of sufficient interest, archaeologists then revisit the site via either divers, remotely operated vehicles (ROVs) or AUVs, depending on the site's accessibility. If this results in a serious discovery, the site is often mapped with photogrammetry and a three-dimensional reconstruction may be created. The reconstruction allows for continued study of the site by archaeologists, preservation of the site, and access for those who are not qualified technical divers to explore it virtually.

Our original work in [20] proposed a multi-step process to identify and validate these sites. First, an AUV takes high-altitude, low-resolution scans of a large area. Second, the scans are fed into image processing software where potential sites are identified and ranked. Third, a path to visit the highest-ranked sites is planned and an AUV is deployed to take low-altitude, high-resolution images.

The goal of this work is to improve the image processing step in order to classify potential areas of interest more effectively. Specifically, this paper presents a number of contributions to the field of underwater robotics including:

- An archaeological site identification algorithm pipeline that processes AUV-obtained SSS data to yield locations of underwater archaeological sites of interest.
- Demonstrated increases in performance over standard CNN approaches due to the incorporation of conventional feature extraction methods.
- Experimental validation with a series of AUV deployments in Malta that produced actual site identifications.

The paper is organized as follows. Section 2 presents relevant background information on the use of AUVs in shipwreck search and mapping and the application of machine learning to this field. Section 3 gives an overview of the algorithm and an in depth view of each stage of the pipeline. Section 4 details how the data was collected and how the algorithm performed. Finally, Section 5 discusses conclusions and future work.

2 Background

When surveying a large area to search for shipwrecks or other archaeological sites of interest, marine archaeologists often turn to SSS sensors [1]. SSS, an acoustic imaging technique, has a number of advantages over light-based approaches like video cameras. While SSS typically creates lower resolution imagery than cameras, acoustic waves do not attenuate as quickly in water and are able to capture large objects at greater range in a single frame [23].

Once the sonar data has been collected, it can be converted to images which are analyzed by archaeologists for potential sites of interest. In order to identify such regions, the archaeologists look for defined objects in the scans, using common attributes such as the size, shape, and texture of the object as well as the presence or absence of a shadow to differentiate archaeological artifacts from terrain features and noise in the data [20].

A number of different approaches exist for processing this image data automatically. First, conventional image processing techniques allow for the extraction of a wide variety of image features, such as contours, edges, and key points [14]. These features can be obtained efficiently through libraries like OpenCV [3]. Previous work has shown the effectiveness of these techniques on images extracted from SSS data. Chew, Tong, and Chia [5] use features like size and outline regularity to label objects as man-made or not, while Daniel et al. [6] use bounding boxes and object distinctiveness for classification.

However, in recent years, convolutional neural networks have emerged as the technique of choice for image processing tasks. State-of-the-art convolutional neural network models are trained on data sets like Microsoft COCO [12] and ImageNet [19] which contain hundreds of thousands or even millions of images. The vast quantities of data are necessary to properly fit the large number of parameters these models contain ([9], [22]). If too little data is used, the model overfits, meaning that is is accurately able to classify the images used for training but not other examples of the same type.

Unfortunately, this volume of data is not always available, i.e. in marine archaeology where data is expensive and difficult to collect [8]. A number of different approaches exist for dealing with overfitting in a neural network, including batch normalization [10], transfer learning [16], dropout [24], and regularization [25].

Another technique, data augmentation, is often used to ”generate” more data without having to collect more samples. Perez and Wang [15] propose a number of ways to do this, including traditional transformation techniques like shifting, rotating, flipping, shading, etc. and two neural network-based approaches, generative adversarial networks—which start with noise and produce images similar to the training data—and their own augmentation network—which takes two training images and combines them to produce a third.

A number of different machine learning techniques have been applied to the classification of SSS data. Previous work has demonstrated the effectiveness of machine learning for object detection in SSS images ([4], [7], [17]). Neural network-based techniques have also been used with promising results ([11], [13], [21]).

Data augmentation has also been applied to SSS images. In order to augment their data and classify mine-like objects (MLOs), Barngrover et al. [2] drew polygons around each MLO and copied only the relevant pixels within that polygon onto examples of images without any features of interest. They also copied negative image pixels in the same shapes from one scan to another in order to account for any potential artifacts that may have been introduced in the augmentation process. This semi-synthetic data was then used with Haar-like and local binary pattern features as well as boosting to classify windows as containing MLOs or not.

In contrast to the previous works cited above, this work demonstrates how data augmentation and traditional image feature extraction methods can be combined and used to improve the performance of CNNs when applied to SSS data.

3 Intelligent Shipwreck Search

Searching for shipwrecks and other sites of archaeological interest in an unsurveyed area involves multiple rounds of AUV deployments combined with post-deployment image processing and AUV path planning. First, an area is chosen to be surveyed based on historical data. High altitude AUV lawnmower paths are planned to cover the entire area to obtain low-frequency, low-resolution SSS. The sonar scans are then fed into a sonar mapping software that converts the raw data into images to be analyzed.

The images are then examined for potential areas of archaeological interest. Once such regions have been identified, new low-altitude AUV paths are planned to obtain high-frequency, high-resolution sonar and video data. This new data can be used to confirm the type of site (e.g. plane wreck), generate maps for additional AUV path planning, and create three-dimensional wreck models using photogrammetry.

Traditionally, in the site identification stage, an archaeologist manually looks through each sonar image, a process which can be time consuming for large data sets and is largely dependent on the experience and skill of the archaeologist. This paper demonstrates that the same process can be done automatically using a site identification algorithm pipeline that takes as input SSS images and outputs potential sites of interest with their corresponding locations in the images. The next section provides details of this pipeline.

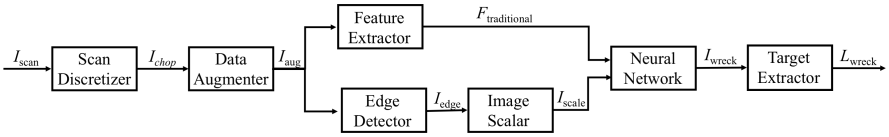


Fig. 2: Block diagram of the sonar scan processing pipeline

3.1 Site Identification Algorithm Pipeline

The pipeline seen in Figure 2 effectively identifies archaeological sites of interest in a SSS image. First, a scan I_{scan} is discretized into N 128x128 pixel sub-images, the set of which is termed I_{chop}. Each of the N sub-images i_{chop} is considered to be an area within which a potential site could be located. As such, each sub-image i_{chop} is passed through a data augmenter, that generates S moderately transformed images of size 200x200 pixels.

Each augmented sub-image, the set of which is termed I_{aug} with cardinality $|I_{aug}| = NS$, is inserted into two different processes, one that extracts image features $F_{traditional}$ with traditional image processing techniques and one that filters the image to detect edges and scales it back to a 128x128 pixel image, the set of which is called I_{scale}. Both $F_{traditional}$ and I_{scale} are passed to a neural network for processing. The neural network labels each of the NS sub-images in I_{scale} as either containing a site of interest or not, and outputs the set of images including such a region I_{wreck}.

Finally, the target extractor function outputs L_{wreck}, which contains the locations of only those sub-images i_{chop} in I_{chop} identified by the target extractor as containing a site of interest. Details of each step in the pipeline are provided below.

3.2 Scan Discretizer

When given a new SSS image, I_{scan}, the algorithm first uniformly and evenly divides the scan into 128x128 pixel sub-images. The size 128x128 pixels was chosen because it is small enough to allow for smaller sites of interest to take up a significant percent of the image, while large enough to allow for most of a larger site to be captured. If the site is too large, data augmentation described in Section 3.3 is sufficient to allow all regions of a partially cut-off area of interest to be identified (see Figure 5b).

Associated with each sub-image i_{chop} is a 4-tuple label that includes the X and Y coordinates of the top left corner as well as the width and height of the area. This label will be outputted at the end of the pipeline and can be used to help geo-localize the sub-image if it is identified as containing a site of interest.

3.3 Data Augmenter

Data augmentation serves a number of purposes within the proposed algorithm. First, it allows for the creation of a larger data set for training purposes, the details of which are described in Section 4.3. Second, augmenting the data inputs to the trained network (e.g. at testing time) increases the likelihood it will be correctly classified by the pipeline.

Each sub-image i_{chop} is passed through data augmentation S times to create S augmented versions of i_{chop}. Transformations are used to create the augmented versions of the sub-images, specifically scaling, translating, flipping and shearing. Details of these transformations are provided below.

The first transformations are **scaling** and **translating** which are applied in conjunction with each other. First, a value for the scaling factor γ is selected by randomly sampling a log-uniform value between $\frac{1}{2}$ and 2. Second, the values for the width and height of a sub-image are calculated such that when later scaled by γ will yield a size of 200x200. Next, the coordinates of the top left corner of the sub-image to be extracted from the full scan are randomly selected such that the resulting box defined by the top-left corner, width, and height will encapsulate the entire region i_{chop}. Finally, the sub-image is extracted from the full scan and scaled by γ to a size of 200x200 pixels.

Two types of **flipping** may be applied to each sub-image, a horizontal flip and a vertical flip. This is highly relevant for SSS images because features protruding from the sea floor block the sonar waves, creating an ”acoustic shadow” that extends away from the nadir, the area directly below the sensor. Therefore, features on the left side of a scan image extend shadows to the left, and features on the right side of the image extend shadows to the right, (see Figures 5a and 5b). In order to remove any correlation between the side of the scan and the label, all regions on the right side of the nadir of the scan are flipped horizontally. Additionally, to further augment the data, each image has a 50% chance of being flipped vertically.

Finally, in order to further distort the sub-image, each one is either horizontally or vertically **sheared**. The result of OpenCV’s warpAffine, which is used for this task, is a parallelogram. In order to maintain consistent sub-image dimensions (200x200), the triangles on each side of the parallelogram are ignored and only the middle 200x200 square is passed to the next stage.

3.4 Image Feature Extractor

Seven additional features are extracted from each sub-image in I_{aug} by using standard vision processing techniques in order to augment the information extracted by the neural network. The seven features included are the number of corners, number of lines, number of blobs, number of ORBs, number of contours, minimum pixel intensity value, and maximum pixel intensity value. Most of the features are extracted using functions provided in the OpenCV Python library.

Corners are found using Harris corner detection which uses a windowing function multiplied with the x and y gradients of the image to determine which pixel regions contain corners. Lines are detected using Canny edge detection, an algorithm that takes the gradient of an image, thresholds on intensity, and classifies the strength of the edges. Contours are found by finding connected boundaries with the same pixel color or intensity. Blob detection operates by determining contours and filtering them by certain features, like circularity and convexity [14]. Oriented FAST and Rotated BRIEF (ORB) is a feature detector developed by OpenCV that picks out important features for use in applications like panorama stitching [18].

Each of these features is extracted with a specific goal in mind. Corners and lines highlight wrecks, since man-made objects are almost always made with defined lines and corners, but natural features having gone through much more erosion and weathering, have smoothed out their sharp edges. Blobs help identify rocks, since as mentioned above, natural objects are often eroded and formed into rounded objects. ORBs and contours measure the number of important features in each image. Because archaeological sites of interest stand out from the background, they are most often detected in positive images. The minimum and maximum pixel intensities highlight the presence of a large object that casts a large shadow.

Due to high levels of noise in the images generated from the sonar data, all sub-images were first blurred with Gaussian and median blurring before attempting to detect the seven features. This makes it difficult to detect smaller objects, but vastly reduced the number of false positive identifications by the feature extraction.

The output of the image feature extractor is the feature set $F_{traditional}$, which includes values for the seven features associated with each of the NS sub-images.

3.5 Edge Detector

Using edge detection on the sonar scan sub-images dramatically improves site identification performance. In this stage of the pipeline, each of the NS sub-images in I_{aug} are passed through median and Gaussian blurring filters, before being passed through a Canny edge detection algorithm provided in OpenCV. The output of this stage is a new set of sub-images I_{edge}.

3.6 Image Scalar

The sub-images in I_{edge} are of dimension 200x200 pixels. This size was found to be effective for feature extraction, but too large for practical neural network training duration. To reduce training time without loss of neural network performance in terms of precision and recall, each sub-image is resized to be 128x128 pixels and outputted in the sub-image set I_{scale}.

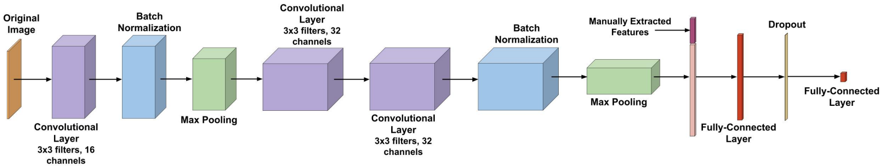

Original Image
Convolutional Layer 3x3 filters, 16 channels
Batch Normalization
Max Pooling
Convolutional Layer 3x3 filters, 32 channels
Convolutional Layer 3x3 filters, 32 channels
Batch Normalization
Max Pooling
Manually Extracted Features
Fully-Connected Layer
Dropout
Fully-Connected Layer

Fig. 3: Neural network architecture used in the pipeline

3.7 *Neural Network*

Each sub-image in I_{scale} and the corresponding feature set $F_{traditional}$ are then used as an input to the neural network (NN). The architecture for the network is shown in Figure 3, and is based on an architecture proposed by Perez and Wang [15]. Though several modifications to this architecture were explored and implemented, no difference in performance was observed.

The first series of steps in the NN architecture form a convolutional neural network (CNN):

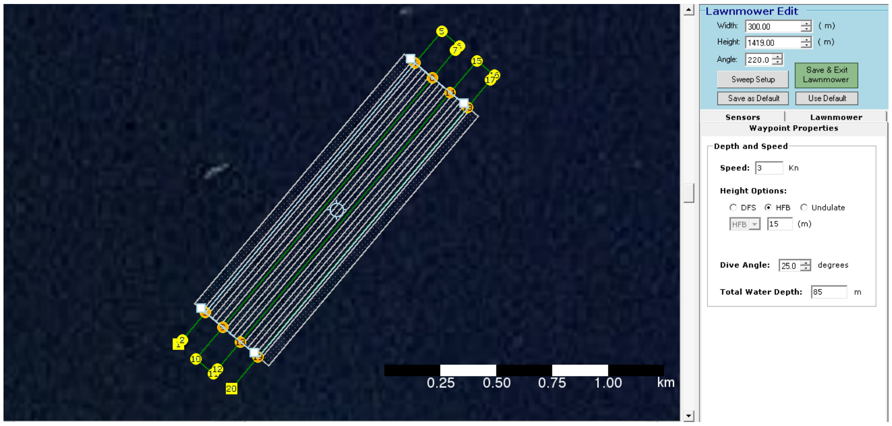
1. Convolutional layer with 3x3 filters and 16 channels. ReLU activation.
2. Batch normalization.
3. Max pooling with 2x2 filters and 2x2 stride.
4. Convolutional layer with 3x3 filters and 32 channels. ReLU activation.
5. Convolutional layer with 3x3 filters and 32 channels. ReLU activation.
6. Batch normalization.
7. Max pooling with 2x2 filters and 2x2 stride.

In order to combat overfitting in the network, regularization is also applied to each of the convolutional layers. The output of the final max pooling layer is then flattened in preparation for the next stage of the pipeline.

The flattened output of the convolutional neural network is then concatenated with the output of the image feature extractor and fed into the following architecture:

1. Fully connected layer with output dimension 1024. ReLU activation.
2. Dropout.
3. Fully connected layer with output dimension 2.

The final output of the last step of the NN is a label for each of the NS sub-images in I_{scale}, i.e. whether the sub-image contains a site or not. Hence the NN function block outputs I_{wreck}, the subset of the NS images that were labeled as containing a site of interest.



Width: 300.00 (m)
Height: 1419.00 (m)
Angle: 220.0

Speed: 3 Kn

Height Options:
○ DFS ● HFB ○ Undulate
HFB 15 (m)

Dive Angle: 25.0 degrees

Total Water Depth: 85 m

Fig. 4: Screenshot from VectorMap Software

3.8 Scan Target Extractor

The target extractor function outputs L_{wreck}, which contains the locations of sub-images i_{chop} in I_{chop} that contain a site of interest. A sub-image i_{chop}'s location is included in L_{wreck} if some minimum fraction of the S sub-images generated by augmenting i_{chop} are labeled by the neural network as containing a site of interest. In general, the threshold fraction is $\frac{S}{2}$.

4 Experiments

Several AUV deployments were conducted during actual archaeological site search expeditions in the Mediterranean Sea in which SSS data was collected and used to validate the performance of the site identification algorithm pipeline.

4.1 Data Collection

Deployments were conducted with a BF12-1006 Bluefin AUV equipped with an Edgetech 2205 Dual Frequency 600/1600 kHz SSS module. High altitude lawnmower pattern paths, e.g. shown in Figure 4, were executed for each deployment. Two specific data sets were collected, using the same AUV but taken a year apart, (i.e. 2015 and 2016), and labeled throughout the rest of the paper as data set A and data set B. The survey areas for these data sets were 4 km by 4 km. Example annotated images of the SSS data collected is shown in Figures 5a and 5b.

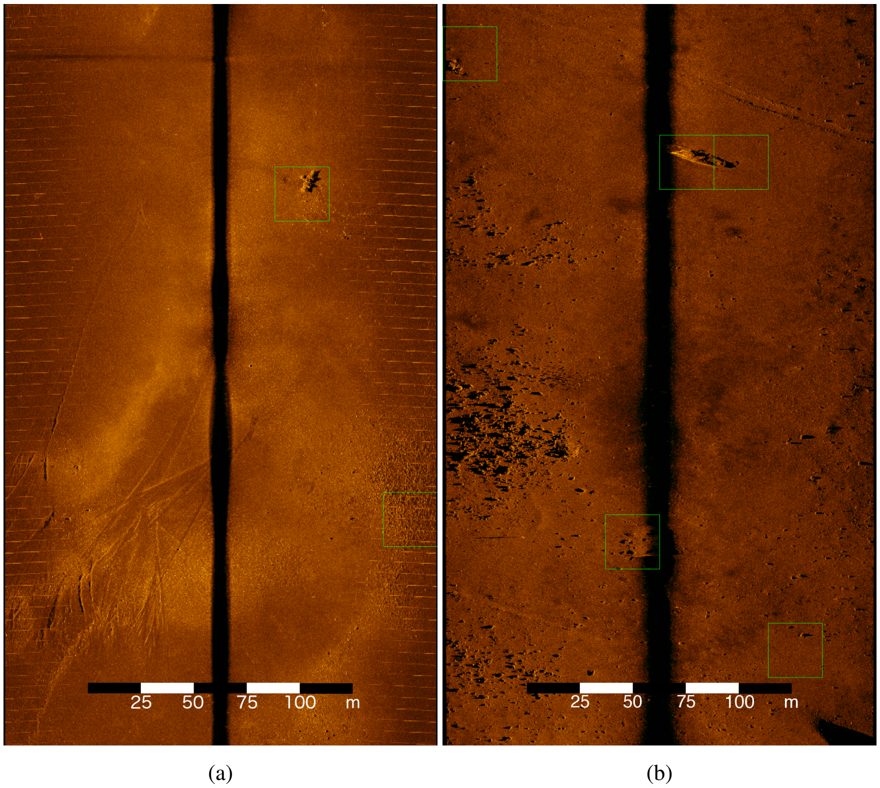


Fig. 5: Examples of scans taken from the (a) A and (b) B data sets labeled with predictions from the site identification algorithm. Note that each image corresponds to only a fraction of the length of one path line from a lawnmower pattern such as the one shown in Figure 4). Images courtesy of Vulcan Inc./University of Malta.

4.2 Labeled Data Set Creation

The data used in the following experiments was labeled by both an experienced archaeologist and student researchers. Labeling areas of archaeological interest is extremely difficult, even for experienced humans. For example, a recent data set collected by the team in 2017, was labeled by two different sets of researchers. Only 26% of the proposed sites were proposed by both groups.

This discrepancy is due to a number of difficulties associated with identifying regions of interest. First, the resolution of the scans is about 19.5 cm^2/pixel, making fine details challenging to make out. In addition, noise generated in the process of taking sonar data and converting it to an image can create artifacts that look like sites of interest or obscure the underlying data. Finally, debris, rocks, and other features of the natural terrain contain many of the same features as objects of interest.

Unfortunately, these features are not uniform across data sets and can depend on a number of factors, including the sonar sensors used, the area being surveyed and the ocean conditions when the data was taken. Notice that Figure 5a, from data set A, contains regular linear artifacts of the sonar scanning process, while Figure 5b, from data set B, features a rocky terrain.

4.3 Neural Network Training

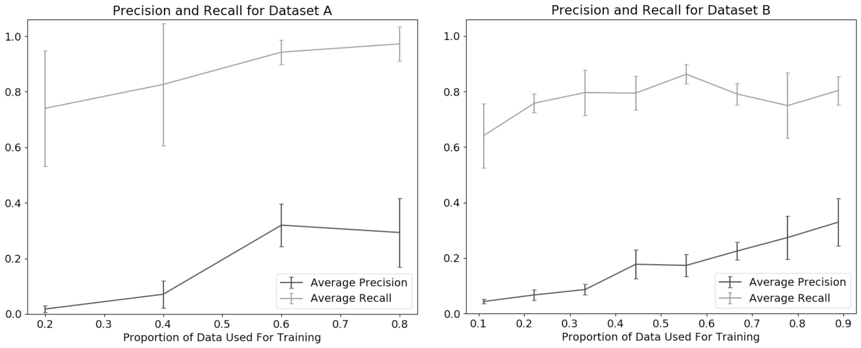
A number of strategies were employed to effectively train the neural networks. First, regions within the scans were selected for training. These included all areas labeled as containing a site of interest as well as a number of hand-picked areas with no sites. These negative regions were selected based on whether or not they contained a feature of the terrain that had not yet been included, as well as whether or not similar regions had previously been consistently misclassified.

Since the results of one trial were used to inform the data used for the next, the images used for validation were randomly selected each time. This minimized bias in the training data by forcing the labelers to consider what data needed to be included to represent as many regions of as many scans as possible.

However, the total number of images in the data set was far lower than is necessary for training the above-described CNN. With the goal of creating a larger data set, each image was passed through the data augmenter. Because of the lack of examples of archaeological sites of interest, the training data set generation process led to the selection of far more regions without sites of interest. To address this, the number of augmented copies of each negative image was reduced so that the total number of positive and negative examples fed to the neural network was balanced. Augmenting both the images containing objects of archaeological value and those that did not had the added effect of reducing any potential bias correlated to features of the augmented image that may have been introduced during the augmentation process. For dataset A, this resulted in augmented 1124 images used for training and validation, while for dataset B, augmented 6024 images were used.

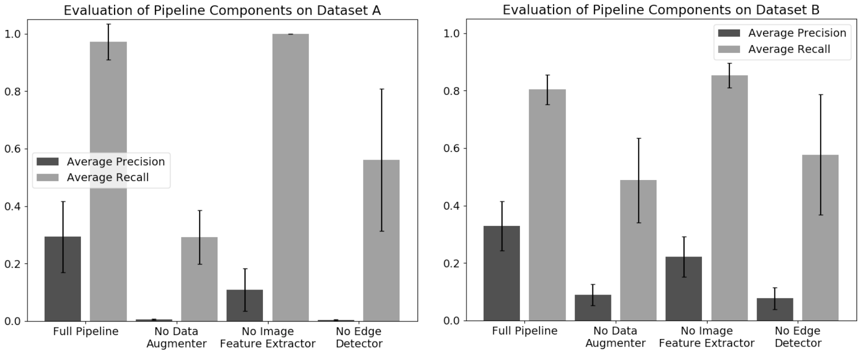
4.4 Results

This section presents the site identification algorithm pipeline performance evaluation experiments. While pipeline performance was measured using precision and recall, two sets of results are presented that ground these measurements in our problem domain. For each trial, a proportion of the sonar scans were selected for training versus validation. Once the network was trained, each validation image was put through the pipeline and the number of true positives, false positives, and false negatives was counted. Because the scans used for validation were selected randomly and the sites of interest are not evenly distributed across the scans, these three num-



(a) (b)

Fig. 6: Validation precision and recall as a function of the proportion of (a) data set A or (b) data set B used for training



(a) (b)

Fig. 7: Validation precision and recall for the algorithm with different blocks of the pipeline removed for (a) data set A and (b) data set B

bers were aggregated across five different rounds to compute precision and recall for a single trial of the algorithm. Six trials of each configuration of the algorithm were averaged to compute the results seen in Figures 6 and 7.

First, precision and recall were measured for cases of increasing fractions of a data set being used for training versus testing. This provides a measure of one's ability to start an AUV search expedition in a new area, and allow training to be done on some fraction of initial images before being confident the pipeline will be successful in identifying sites on the remaining images. Figure 6 illustrates pipeline performance for increasing fractions of training data. As shown in the figures, for data set A, the precision reached 29.34% and the recall reached 97.22%, while for data set B, the precision reached 32.95% and the recall reached 80.39%.

Second, precision and recall were measured for cases with and without some stages of the pipeline to highlight the individual component effects. These components—the Data Augmenter, the Image Feature Extractor, and the Edge Detector—were removed from the pipeline. The network was retrained 80% and 88.89% of datasets A and B respectively. The bar graphs in Figure 7 show the precision and recall for these different configurations. The above figures demonstrate how crucial all three of these stages are in improving the algorithm performance.

Precision and recall are both important values to consider when evaluating the effectiveness of the algorithm. A high recall means that an archaeologist is less likely to miss a potential region of interest, while a high precision reduces the number of sites an archaeologist must revisit. An algorithm that performs well when assessed by both metrics allows archaeologists to explore more sites that could lead to discoveries while spending less time at sites that will not. Though our algorithm exhibits a low precision, it filters out 99.79% of the area in data set A and 99.31% of the area in data set B, highlighting the efficiency of using such an algorithm.

5 Conclusions and Future Work

This work presents a pipeline that identifies archaeological sites of interest in SSS data obtained with an AUV. The proposed algorithm presents data augmentation and image feature extraction as two approaches to increase performance over standard CNN approaches. In addition, the paper provides experimental validation in the form of AUV deployments off the coast of Malta that led to the discovery of sites of archaeological significance.

The algorithm presented achieves a precision and recall of 29.34% and 97.22% on one dataset and 32.95% and 80.39% on the second. These results demonstrate that the proposed pipeline identifies a majority of the archaeological sites tested, but misclassifies some noise and features of the natural terrain as regions of interest. Due to the low precision, this algorithm should be used to inform archaeologists about areas to consider revisiting by giving them a smaller sample size to review.

This paper discusses an automated pipeline for identifying sites of archaeological interest from SSS images. Future work may include testing the algorithm on a wider variety of data, including scans from different altitudes and different angles. Because this data is expensive to collect, another area for future work is simulating the creation of these scans in software. In addition, the pipeline presented can be integrated into the AUV intelligent search system.

Acknowledgements We would like to thank Jeffrey Rutledge, Samantha Ting, Russell Bingham, Andrew Pham, Kolton Yager, Bonita Galvan, and Mitchell Keller for helping us deploy the robot. This work was performed in part at the Claremont Colleges Robert J. Bernard Biological Field Station. This material is based upon work supported by the National Science Foundation under Grant No. 1460153.

References

1. Atallah L, Shang C, Bates R (2005) Object detection at different resolution in archaeological side-scan sonar images. Eur Ocean 2005, doi: 10.1109/OCEANSE.2005.1511727
2. Barngrover C, Kastner R, Belongie S (2015) Semisynthetic versus real-world sonar training data for the classification of mine-like objects. J Ocean Eng 40:48-56
3. Bradski G, Kaehler A (2008) Learning opencv, 1st edn. O'Reilly, California
4. Chang R, Wang Y, Hou J et al (2016) Underwater object detection with efficient shadow-removal for side scan sonar images. OCEANS 2016 - Shanghai, doi: 10.1109/OCEANSAP.2016.7485696
5. Chew A, Tong P, Chia C (2007) Automatic detection and classification of man-made targets in side scan sonar images. Symp on Underwater Tech doi: 10.1109/UT.2007.370841
6. Daniel S, Le Léannec F, Roux C et al (1998) Side-scan sonar image matching. J Ocean Eng 23:245-259
7. Dura E, Zhang Y, Liao X et al (2005) Active learning for detection of mine-like objects in side-scan sonar imagery. J Ocean Eng 30:360-371
8. Gambin T (2014) Side scan sonar and the management of underwater cultural heritage. In: Formosa S (ed) Future preparedness: thematic and spatial issues for the environment and sustainability. Msida, Malta
9. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. arXiv:1512.03385
10. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167
11. Langner F, Knauer C, Jans W et al (2009) Side scan sonar image resolution and automatic object detection, classification and identification. OCEANS 2009-EUROPE, doi: 10.1109/OCEANSE.2009.5278183
12. Lin TY, Maire M, Belongie S et al (2014) Microsoft COCO: common objects in context. arXiv:1405.0312
13. McKay J, Gerg I, Monga V et al (2017) What's mine is yours: pretrained CNNs for limited training sonar ATR. arXiv:1706.09858
14. Nixon M, Aguado A (2012) Feature Extraction and Image Processing for Computer Vision. Academic Press; 3 edition
15. Perez L, Wang J (2017) The effectiveness of data augmentation in image classification using deep learning. arXiv:1712.04621
16. Razavian AS, Azizpour H, Sullivan J et al (2014) CNN features off-the-shelf: an astounding baseline for recognition. arXiv:1403.6382
17. Reed S, Petillot Y, Bell, J (2003) An automatic approach to the detection and extraction of mine features in sidescan sonar. J Ocean Eng 28:95-105
18. Rublee E, Rabaud V, Konolige K et al (2011) ORB: an efficient alternative to SIFT or SURF. Int Conf Comput Vis, doi:10.1.1.370.4395
19. Russakovsky O, Deng J, Su H et al (2015) ImageNet large scale visual recognition challenge. arXiv:1409.0575
20. Rutledge J, Yuan W, Wu J et al (2018) Intelligent shipwreck search using autonomous underwater vehicles. Int Conf Robot Autom, doi: 10.1109/ICRA.2018.8460548
21. Shang C, Brown K (1992) Feature-based texture classification of side-scan sonar images using a neural network approach. Electron Lett 28:2165-2167
22. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
23. Singh H, Adams J, Mindell D et al (2000). Imaging Underwater for Archaeology. J Field Archaeol 27:319-328
24. Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:19291958
25. Wang B, Klabjan D (2017) Regularization for unsupervised deep neural nets. arXiv:1608.04426