A UNIFIED SEMANTIC SEGMENTATION AND OBJECT DETECTION FRAMEWORK FOR SYNTHETIC APERTURE SONAR IMAGERY

Shannon-Morgan Steele

Kraken Robotics

ABSTRACT

Manually identifying objects in synthetic aperture sonar (SAS) imagery is costly and time consuming, making identification through computer vision and deep learning techniques an appealing alternative. Depending on the application, a generalized map (semantic segmentation) and/or a characterization of each individual object (object detection) may be desired. Here, we demonstrate a framework that allows us to simultaneously generate both semantic segmentation maps and object detections with a single deep learning model by chaining together a U-Net model with k-means clustering and connected components. This framework streamlines the model training phase by allowing us to utilize a set of semantically segmented training data to yield both semantic segmentation and bounding box predictions. We demonstrate that the deep learning model can achieve accurate predictions with a small training set through transfer learning from a convolutional neural network pretrained on optical imagery. Results from this unified framework will be presented on images of boulders collected during various surveys using a Kraken Robotics miniature SAS (MINSAS).

Index Terms— Synthetic Aperture, Sonar, Object Detection, Seabed, Segmentation

1. INTRODUCTION

Many seabed object detection applications such as mine hunting and geohazard detection require centimeter scale resolution over large survey areas, making SAS an ideal sensor for object detection. This is especially true for boulder detection applications where it has been shown that the resolution of side looking sonar imagery strongly impacts the quantity of boulders that can be detected [1]. Deep learning object detection workflows can be broken down into two concepts termed classification and localization. Convolutional neural network (CNN) based detectors may perform these in one or two stages. The primary limitations of the two-stage approach are its computation expense and often imprecise bounding boxes that may contain more than one object, especially if the objects are overlapping or small [2]. The one stage approach has more precise bounding boxes, but it often misses smaller objects and thus detects fewer objects overall [2]. Boulder detection in SAS imagery requires the detection of small objects that are often densely populated and overlapping. The size of the objects that are required to be detected in SAS imagery is significantly smaller than most optical imagery object detection tasks. The COCO (Common Objects in Context) benchmark dataset consisting of over 200,000 labeled optical images [3] defines small objects as objects having pixel areas of less than 32^2 pixels [4], since the COCO image size is 640 x 480 pixels, this is equivalent to an object taking up 0.33 % of the image pixels. For comparison, the average area of boulders detected in our SAS imagery is $3 \times 10^{-6}\%$ of pixels. Thus, existing approaches to object detection with CNNs may not be optimal for boulder detection in SAS imagery. Here, we propose and evaluate a new approach to object detection that utilizes deep learning to perform semantic segmentation and computer vision techniques to select object bounding boxes based on the semantic segmentation output. The primary advantages of this approach are:

- Allows for object detection capabilities without object detection training data, which is extremely labor intensive in SAS imagery.
- More interpratable bounding box output. Deep learning approaches are black boxes and thus we are not able to monitor or modify their outputs. However, computer vision techniques provide more control opportunities.
- From a single CNN we can produce an overview map indicating regions that contain boulders as well as detections of individual objects. These can be combined to generate summary maps of the boulder size distribution and other statistics of regions with boulders.

This paper has two novel contributions: 1) the novelty of the framework itself (including a new technique for calibrating k-means centroids across many images), and 2) while there is previous work applying transfer learning from optical imagery to SAS imagery for mine detection [5, 6], to the author's knowledge this is the first application for geohazard detection.

2. METHODS

Our new proposed framework can be broken down into two major steps: semantic segmentation using deep learning and

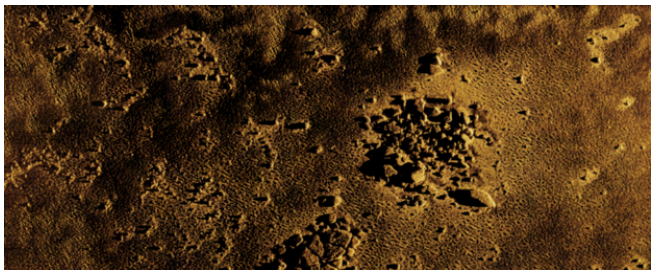
object detection with computer vision. The first task of the deep learning framework was to generate training data using the "lazy labelling" technique, where polygons were drawn around individual boulders. When boulders were overlapping or adjacent to one another, a single polygon was drawn around that region. This training data was then fed into a convolutional neural network (CNN) designed to produce segmentation maps by labelling the pixels as either target (boulders present) or background (no boulders present). The output prediction map does not discriminate between adjacent rocks or the boulder shadows. Thus, k-means clustering was utilized to discriminate between adjacent boulders and their shadows to obtain an improved segmentation map. This improved segmentation map is generally sufficient to produce individual boulder detections using connected components.

2.1. Semantic Segmentation

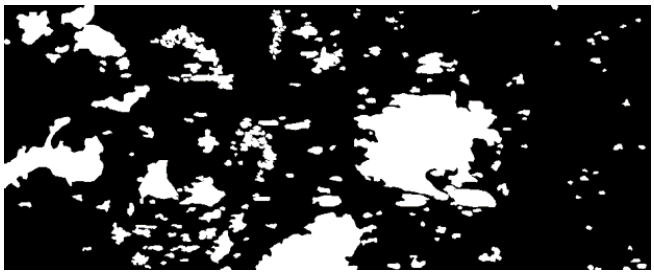
Semantic segmentation typically requires some form of fully convolutional neural network (FCN) to be able to localize features within the image. Here, we have chosen to follow the U-Net architecture [7], as it has been shown to produce accurate predictions, even with a small training set. The architecture of U-Net follows a u-shape consisting of a contracting and an expansive path. The contracting path is a typical CNN while the expansive path utilizes a sequence of up-convolutions and concatenations to increase the resolution of the output. Due to the limited size of our training data we utilize a pre-trained CNN for the contracting path instead of training our own CNN from scratch. We chose the VGG16 architecture pre-trained on ImageNet [8] data as our base CNN. The VGG16 network has been trained to detect objects belonging to 1000 different classes. The filters this network has generated can be modified to detect boulders through the fine tuning transfer learning technique. To fine tune the model we freeze some layers such that their weights don't change and for the rest we utilize the weights as a starting point and the model adjusts the weights based on our training data. We found the model performed best when all the convolutional layers of the VGG16 network were frozen except the last two.

2.2. Object Detection

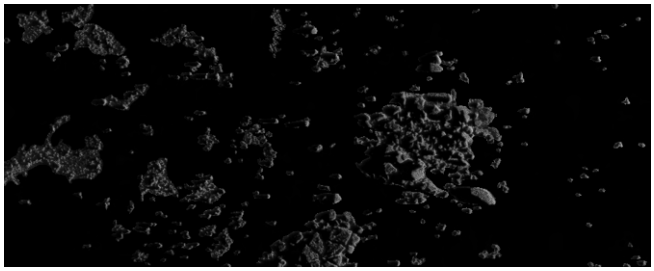
To detect individual targets identified in the semantic segmentation maps we utilize a computer vision technique known as connected components [9]. Connected components is a method for detecting objects in binary imagery. The algorithm picks out individual objects based on the target class pixel connectivity and assigns each connected component a label. Here, we use a connectivity of 8, meaning the target pixels are considered connected if their edges or corners touch. The algorithm then iterates through 2 steps. First, it searches for the next unlabeled target pixel, p. It then utilizes a flood-fill algorithm to label all the target pixels in the connected component. These steps are repeated until all the tar-



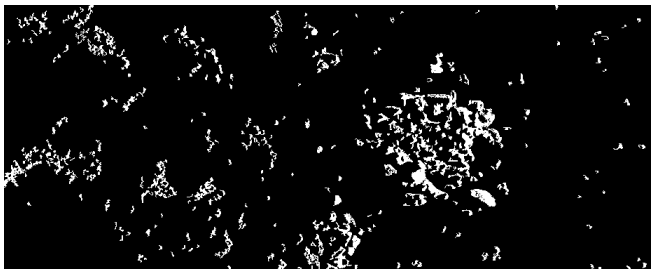
(a) SAS image input to CNN.



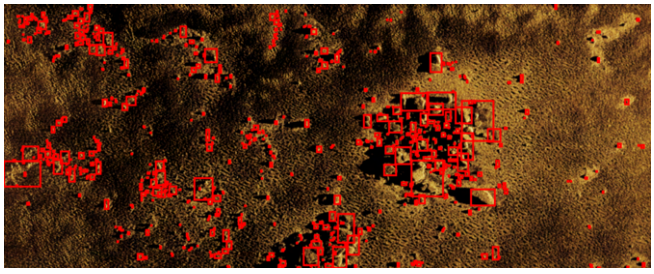
(b) CNN predicted segmentation map for the image in Fig. 1a. White is the target class and black is the background.



(c) Grey scale k-means input image with background pixels in Fig. 1b set to an intensity of zero.



(d) Labels for Fig. 1c determined by k-means clustering.



(e) Bounding boxes identified by the connected components algorithm using the labels from Fig. 1d.

Fig. 1: Overview of method to produce object detections from semantic segmentation maps.

get pixels are labeled. The connected components algorithm yields a list of pixels included in each target, which can then be used to characterize and draw a bounding box around each object.

If we feed our semantic segmentation maps directly into the connected components algorithm the results will be suboptimal in areas where boulders are densely packed because the adjacent boulders will look like one object when viewed as a binary mask. Thus, we need a method for removing the shadows and gaps between boulders that have been labeled as the target class. We can do this using k-means clustering. The k-means algorithm is not sophisticated enough to identify boulders from other seabed features using the raw imagery, largely due to the two classes overlapping in feature space with no obvious clusters. Instead, we can utilize the CNN semantic segmentation output to simplify the problem to one suitable for k-means. A new image is constructed by converting the SAS image (Fig. 1a) to grey scale and setting all the pixels labeled as background in the segmentation map (Fig. 1b) to zero (Fig. 1c). This decreases the complexity of the problem by removing the background seabed texture and features, reducing the k-means segmentation task to separating the low intensity background pixels and the higher intensity target pixels. In this scenario the k-means algorithm is effective at labeling our target class because the two classes form two distinct clusters in feature space. Using the segmentation map output from k-means clustering (Fig. 1d) the connected components algorithm can identify a majority of the boulders individually (Fig. 1e). In imagery with numerous boulders we can end up with large patches of seabed labeled as target pixels (Fig. 2b). In this case, image intensity alone is not sufficient for differentiating boulders and background. We can easily avoid this by introducing a texture measurement as an additional feature in the k-means algorithm (Fig. 2c). Here, we use local range, computed in 3x3 pixel windows.

Various different seabed types can have drastic differences in intensity. If this variation is significant within a single image, the k-means segmentation map and resulting bounding boxes may not appropriately separate target and background pixels, resulting in errors such as the large bounding box drawn around the higher intensity seabed sediment observed in the top image of Fig. 3. This can be avoided by pregenerating the k-means centroids through a technique we've termed k-means calibration. The objective of k-means calibration is to choose target and background centroids in feature space that will represent the entire data set. For computationally efficiency, we randomly select across-track strips of data from different imagery. We start with a single strip from a single image and run k-means to generate the initial centroids. We then iteratively add randomly selected strips to the k-means input feature vector until the target and background centroids converge. These centroids can then either be used directly to produce the final segmentation maps (without having to run k-means on each individual image). As observed in the bottom image of Fig. 3, with k-means calibration we see a drastic improvement in the number of targets detected, especially in the high intensity region (left side) of the image.

For model performance assessment, object detection networks generally return a confidence score for each bounding box. The confidence score (C) is estimated as

$$C = Pr * IoU \tag{1}$$

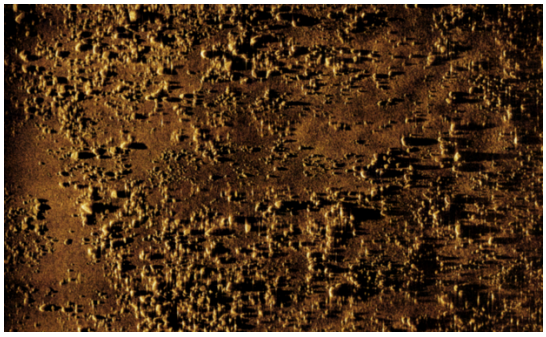
where Pr is the probability of the detected object's class and IoU is the intersection over union between the ground truth and predicted bounding box. We do not have a probability assigned to each bounding box but we do have probabilities assigned to each pixel. We estimate Pr to be the mean probability of the target class pixels contained within a given bounding box. Manually labeled (ground truth) bounding boxes were not available to us, so instead we utilize the technique described above to generate bounding boxes from the validation set ground truth segmentation maps. We also utilize these bounding boxes to quantitatively estimate the object detection model performance through the precision-recall (PR) curve. A well performing model will maximize the area under the PR curve (AUC). The AUC can be used to estimate the average precision (AP) of the object detection model. We follow the PASCAL VOC challenge guidelines [10] to calculate an AP for our model.

2.3. Data collection and processing

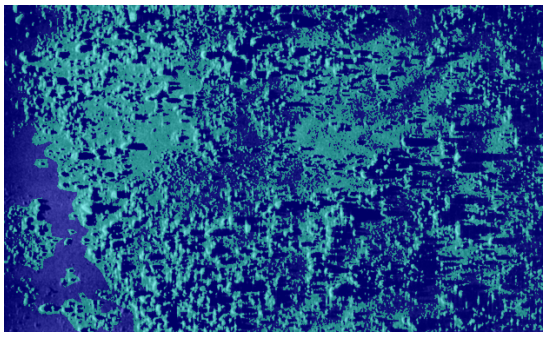
A dataset of 338 labeled SAS images from various MINSAS surveys was generated. We split the labeled dataset into a training set of 287 images and a validation set of 51 images. Since the validation set is not used in the model training in any way (including hyperparameter tuning and early stopping), we will also utilize it to evaluate the model performance after training. Speckle in the SAS imagery is reduced with a gaussian blur. Following this we downsample the images by a factor of four as the SAS imagery is quite large (on the order of 5 million pixels per image). SAS imagery is also variable in size, and thus all the images must be resampled to be the same size for input into the CNN. We have chosen an image size of 224 x 448, as this size should be sufficient to preserve the aspect ratio of the SAS imagery. However, once the model is trained, we make predictions on the variably sized gaussian blurred and downsampled version of the images.

3. RESULTS AND DISCUSSION

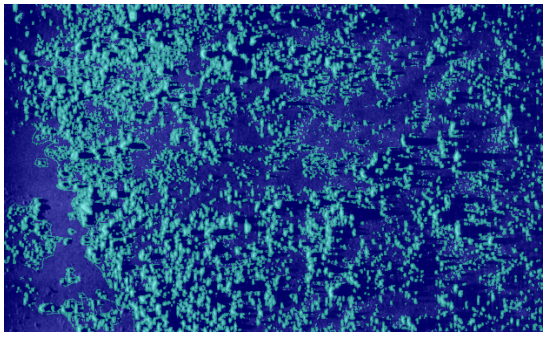
The CNN model was trained with early stopping to prevent overfitting. Overall, the model yields satisfactory predictions; the model can achieve a Dice coefficient of 0.96 on the validation set. For the bounding box prediction performance we present two different PR curves based on different segmentation target class probability (P) cut-offs: 1) label pixels with $P > 0.5$ as the target class and, 2) label pixels with $P > 0.3$



(a) SAS image input to CNN.



(b) Fig. 2a overlayed with labels determined by k-means clustering with only image intensity as a feature. Blue is background class, green is the target class.



(c) Fig. 2a overlayed with labels determined by k-means clustering using intensity and texture features. Blue is background class, green is the target class.

Fig. 2: Including intensity and texture features in k-means can allow individual boulders to be detected in high density areas.

as the target class (Fig. 4). A no-skill classifier PR curve would be a horizontal line. The model achieves an AP of 0.67 and 0.54 for the $P > 0.3$ and $P > 0.5$ cut-offs, respectively, indicating that P must be chosen to meet the desired performance. If we use $P > 0.5$ the model will have high precision and thus it will return few false positives but will suffer in recall. If $P > 0.3$ the model will identify more targets (higher recall), but it will also yield more false positives (decreased precision).

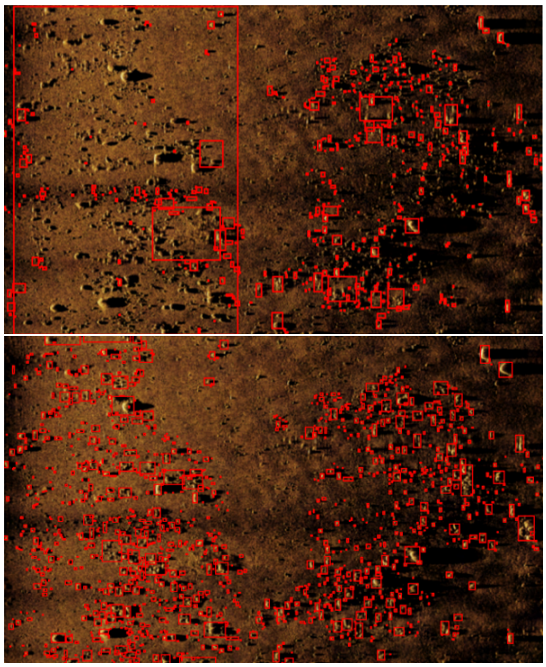


Fig. 3: Top: bounding boxes generated from k-means segmentation map without calibration. Bottom: bounding boxes generated from k-means segmentation map with calibration

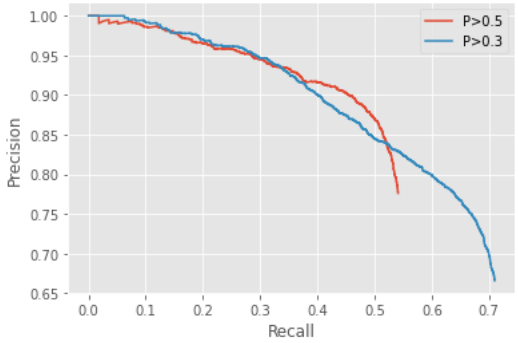


Fig. 4: PR curve for object detection on validation data.

4. CONCLUSION

Here, we presented a novel technique to obtain semantic segmentation maps and object defections from a single CNN. The framework relies on computer vision techniques to derive object detections from semantic segmentation maps. There are many benefits of this unified framework, including easier ground truth labeling and the capability to provide object detection summary maps with corresponding details (such as boulder size distribution). Using transfer learning and a small number of training images we can produce semantic segmentation maps that achieve a Dice coefficient of 0.96 on validation data. We can achieve an AP of up to 67 % for object detections derived from the predicted segmentation maps.

5. REFERENCES

[1] Gitta von Rönn, Klaus Schwarzer, Hans-Christian Reimers, and Christian Winter, "Limitations of boulder detection in shallow water habitats using high-resolution sidescan sonar images," *Geosciences*, vol. 9, no. 9, pp. 390, 2019.

[2] Khaoula Drid, Mebarka Allaoui, and Mohammed Lamine Kherfi, "Object detector combination for increasing accuracy and detecting more overlapping objects," *Image and Signal Processing: 9th International Conference, ICISP 2020*, June 2020.

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft coco: Common objects in context," *Computer Vision – ECCV 2014*, p. 740–755, 2014.

[4] "Microsoft coco: Common objects in context, metrics," *https://cocodataset.org/detection-eval*.

[5] N Warakagoda and Ø Midtgaard, "Transfer learning with deep neural networks for mine recognition in sonar images," *Proceedings of the International Conference on Synthetic Aperture Radar and Sonar, Lerici, Italy*, 2018.

[6] N Warakagoda and Ø Midtgaard, "Retrieval of similar targets in synthetic aperture sonar images with deep learning," *Proceedings of the International Conference on Underwater Acoustics (ICUA)*, 2020.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," *International Conference on Medical image computing and computer-assisted intervention-MICCAI 2015*, 2015.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[9] Rafael C Gonzalez, *Digital Image Processing*, Pearson Education India, 2009.

[10] Mark Everingham, Luc Van Gool, Christopher K. Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2009.