This WACV 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Instance Segmentation of Benthic Scale Worms at a Hydrothermal Site

Bhuvan Malladihalli Shashidhara
University of Washington
Applied Physics Laboratory
msbhuvan@uw.edu

Mitchell Scott
University of Washington
Applied Physics Laboratory
miscott@uw.edu

Aaron Marburg
University of Washington
Applied Physics Laboratory
amarburg@uw.edu

Abstract

Subsea hydrothermal vents, typically existing at water depths below natural light penetration, contain diverse and unique macrofaunal environments. Traditionally, long-term ecological observation has been difficult as the extreme depth, temperature and pressure make in situ video surveys challenging. However, the introduction of subsea cabled arrays has allowed for the long time series collection of high definition imagery from these vents. To study the benthic hydrothermal vent environment, we propose an inference pipeline consisting of a U-Net followed by VGG-16 CNN to perform instance segmentation of scale worms, a specific macrofaunal family. The developed pipeline exhibits an average precision (AP) of 0.671 AP@[0.5], despite the difficult camouflaged imagery and low training data inputs. We further explore full pipeline training data requirements, as the dynamic scene in question requires the pipeline to be re-trained on an approximately monthly basis for effective segmentation. We find that the VGG-16 CNN portion of the pipeline is typically more sensitive to training data variation than the U-Net portion.

1. Introduction

The NSF-funded Ocean Observatories Initiative (OOI) Regional Cabled Array (RCA) offers an unprecedented capacity for observation of deep ocean processes [2] [3]. In operation since 2015, the RCA provides power and network connectivity to multiple study sites in the Northeast Pacific, enabling new forms of realtime, data-driven scientific inquiry into subsea geology, chemistry, biology and physical oceanography. A major study site on the RCA is Axial Seamount, an active volcano located on the Juan de Fuca plate spreading center $\sim$ 500 km off the Oregon Coast [21]. Axial caldera, located at the top of the volcano at $\sim$ 1500m water depth, hosts a diversity of scientific instrumentation for studying the tectonic and volcanic processes occurring at Axial and the interaction between those processes and the overlying water column. Axial also hosts a number of

Figure 1: Map of the OOI Regional Cabled Array. Axial seamount, located approximately 500km off the Oregon coast, is at the center of the image. *Image Credit: UW/NSF-OOI.*

sites where heated, mineral-laden seawater is expelled from the seafloor, creating local prominences, or vent chimneys, which host thriving chemotrophic ecosystems. A map of Axial and the RCA is shown in Figure 1.

At one such vent site, the RCA has installed a high-definition camera (*CamHD* in the system nomenclature), which sits approximately 1.5m from *Mushroom*, a 2m tall hydrothermal vent (Figure 2). Eight times per day, CamHD conducts a $\sim$ 12 minute video survey of Mushroom, panning, tilting and zooming over the side of the vent facing the camera. This HD video is archived within the OOI data repository, with more than 11,000 such video surveys archived at present. During each survey, the camera observes the full extent of the vent (Figure 3), and performs close examinations of a number of regions of interest (Figure 4), capturing a diverse, benthic ecosystem, where tube and palm worms, pycnogonidas, fish, and scale worms thrive in the extreme conditions on the flanks of the vent.

As the camera follows a pre-programmed course, every video shows a consistent set of locations on the vent, opening the possibility of long-time-series studies of geological and biological change at Mushroom. Unfortunately, due to the sheer size of the historical record, the manual extraction of useful quantitative metrics is impractical. Therefore,

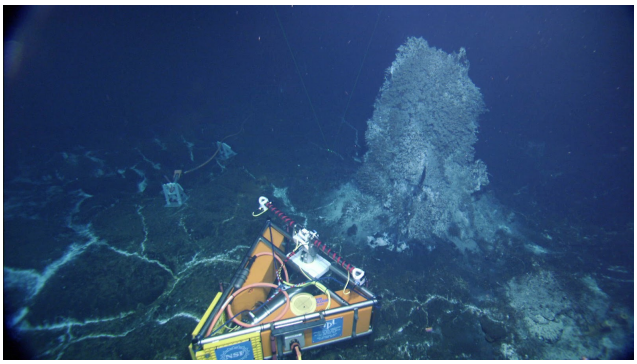

Figure 2: CamHD looking at Mushroom. *Image Credit: Credit: UW/NSF-OOI/CSSF*.

there is a strong incentive for the development of image analysis-based tools for extracting geological and biological metrics from CamHD video and imagery.

As an initial exploration, we investigate the use of machine learning to identify and segment benthic *scale worms* (i.e. Polynoidae, a subsea worm family) [1] from CamHD still images (with two examples shown in Figure 4). We focus on this group of fauna because: 1) they are mobile, and thus capable of migrating in response to local environmental conditions; 2) they are present in relatively large numbers and are common throughout the video record; 3) they have a well-defined, stable outline (unlike, for example, palm worms, which have an indistinct, frond-like outline and are in constant motion); 4) the coloration of many individual scale worms (though not all) appears as a contrasting pink when lit by CamHD's lights – bearing in mind that any light is unnatural in the environment; and 5) as distant relatives of the "pill bug" insects in our gardens, they are relatively charismatic.

The scene in question is difficult to segment for several reasons:

1. The scale worms are well camouflaged against a complex, dynamic background;
2. Schlieren (i.e. *waviness* in imagery due to heat) from hot fluids emerging from nearby vents and cracks can severely distort the image;
3. The absence of labeled training data; and
4. The sequence of images contain both slow change on the scale of days to weeks, due to e.g., the growth of bacterial mats (the white "fluff" shown in Fig 4); as well as sudden, discontinuous changes, for example when individual lights on the system fail.

The latter point is of particular interest, as the imaging conditions at Mushroom evolve continuously throughout the video record. The relationship between the performance of image-segmentation algorithms and the temporal gap between training and realtime data, as well as the computational costs for incremental algorithm retraining, is explored in the presented work.

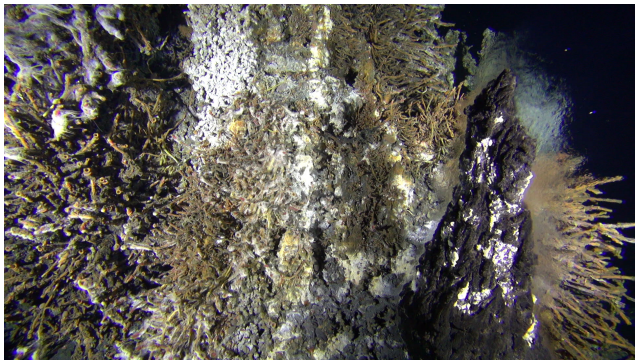

Figure 3: Sample image from CamHD, taken at CamHD's shortest focal length (widest angle). This image shows the significant tube worm populations on Mushroom. It also shows a freshly formed chimney (right) growing from the base of Mushroom where mineral-laden hot water is emerging from the vent in the seafloor. *Image Credit: UW/NSF-OOI*.

In this paper, we develop a hybrid inference pipeline consisting of a U-Net [25] followed by a VGG-16 CNN [27], and evaluate its performance on still frames taken from CamHD video of Mushroom. We show that this model can achieve considerable average precision values, despite the difficult scene and lack of substantial training data. Additionally, we show that the VGG-16 portion of the hybrid pipeline is typically more impacted by additional training data than the U-Net portion, an important attribute for model maintenance given the continuous change in the video content.

We believe our work has two primary benefits: First, we successfully demonstrate the segmentation of a spe-

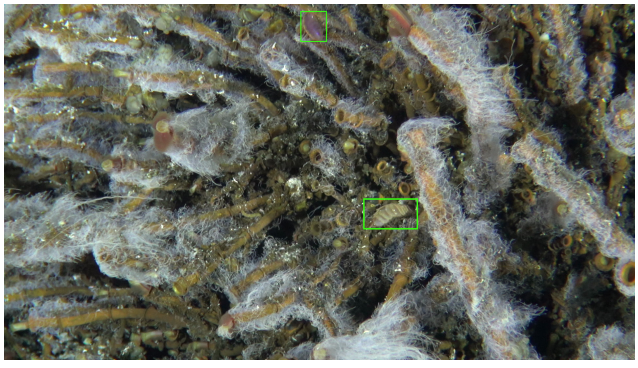

Figure 4: Another sample image from CamHD, taken near CamHD's longest focal length (most zoomed in). This image shows a number of macrofauna, including tube and palm worms, white filamentous bacterial growth, and two scale worms identified by the bounding boxes *Image Credit: UW/NSF-OOI*.

cific macrofauna species from CamHD video, opening the possibility of algorithmically generating quantitative spatial statistics about scale worm distributions throughout the CamHD record. Second, we demonstrate that the presented network architecture is capable of instance segmentation in difficult, dynamic and camouflaged environments.

2. Related Work

The *in situ* study of benthic scale worms is dominated by the cost and difficulty in collecting data. Mushroom, while far from the most inaccessible scale worm habitat, is located at 1500 m water depth, where the ambient pressure is approximately 147 bar (2100 psi). While manned submarines and remotely-operated vehicles (ROVs) have been used for short visits to hydrothermal vent sites, it is only with the introduction of cabled observatories, like the RCA [3] and the similar Ocean Networks Canada (ONC) Neptune ocean observatory [4, 20], that true realtime, long-term observation of benthic habitats has become feasible. Cuvelier *et al*., Lelièvre *et al*., and Lee *et al*., specifically, have studied the behavior and migration patterns of different macrofauna species, including scale worms, on the *Endeavour* and *Lucky Strike* vent fields [14, 5, 15, 6, 16] using conventional uncabled instrumentation and ROV observation. However, the Endeavour vent field is an active study site for the ONC Neptune network, and cabled observations may play a role in future Endeavour observation. While these previous studies do examine the rhythm, dynamics, and temporal macrofauna variation due to temperature and "astronomical and atmospheric" variations, they have not utilized a deep learning image segmentation algorithm to help with continuous data collection under challenging environments. We believe our work could be leveraged to accelerate population assessments such as those presented in this work.

The generalized study of object detection in imagery using deep learning has been extensively studied in recent years [8, 10, 17, e.g.,]. Some examples of attempts of wildlife classification research are as follows. Norouzzadeha *et al*. studied the performance of different networks on the 'Snapshot Serengeti' database, a large multi-million African savanna image database [19, 28, 33]. Shi *et al*. created 'FFDet', a network for the detection of coral reef fish, and compared their network to other state-of-the-art segmentation networks like YOLO [22, 23], SSD [18], and Faster-RCNN [24] [26]. Xu and Matzner segmented fish from image scenes near water power applications (e.g., marine energy converters) using the YOLO network [31]. Although each of these previous architectures are appropriate for their specific application, they have not been tested in the camouflaged environments where our segmentation takes place. Furthermore, the dynamic background scene required network re-training. This, in turn, necessitated the usage of a network with a small training dataset requirement to reduce workload during each re-training cycle. While YOLO did not fit these requirements, *U-Net*, a popular biomedical semantic segmentation network developed by Ronneberger *et al*., works well in camouflaged environments and does not require a large training dataset [25].

3. Proposed Methodology: U-Net and VGG-16 Inference Pipeline

The U-Net architecture utilized here consists of an "up" and "down" path, which forms a "U" shaped structure. The up path i.e., the contraction/encoder step, captures image features and context via a traditional stack of convolution and max pooling layers, while the down path i.e., the symmetric expanding/decoder step, localizes the object [25] [13].

For our network architecture, we added batch-normalization layers after every convolution layer to increase network convergence speed, and to add a regularizing effect [11]. This modified U-Net network used a binary cross-entropy loss with an Adam optimization algorithm at a learning rate of 10^{-3}, and was trained for 300 epochs. Furthermore, our network operates on individual *patches*, i.e., smaller image subsets, and not full CamHD images.

A solo U-Net model, however, is not sufficient for accurate segmentation in this environment as false positives are high (while we do not provide U-Net false positive statistics here, see Figure 7 (b) and (c) for an example). To that end, we add a VGG-16 CNN model which verifies the U-Net segmentation masks.

Specifically, the patch-level outputs from U-Net are stitched to form a whole-image-segmentation-mask, and they undergo probability and area thresholding. Additional morphological operations, e.g., dilation, are applied to remove holes in the mask. Finally, connected components are extracted and considered as candidate regions. Probability and area thresholding removes smaller connected components, and therefore overlapping region proposals are unlikely. A complete description of our inference pipeline can be seen in Figure 5.

In this work, we only analyzed the U-Net + VGG-16 CNN network that we developed, and did not compare performance to other networks like MaskRCNN. We did not conduct this comparison for two primary reasons: 1) the amount of training data was limited, and we did not believe that we had sufficient quantities of examples at the onset of development for a comparison, and 2) we were motivated to develop a more 'modular' network, where we could more finely observe and tune the individual model elements. A formal comparison to other deep learning networks for instance segmentation like MaskRCNN is considered as future work.

3.1. Data Collection, Preprocessing and Training

Scale worm segmentation from Mushroom imagery focuses on three out of the 27 regions visited by CamHD on each pre-programmed video collection sweep. These regions are captured at the camera's greatest level of zoom (longest focal length), and are in regions which frequently host scale worms. From these three scenes, we sampled 352 images from July to October, 2018 (4 months) to serve as the training dataset.

We used the labeling tool *LabelMe* [30] to manually create 537 segmentation masks from the sampled images for our training set. Due to the difficulties in spotting camouflaged scale worms against the vibrant Mushroom background, we utilized a *sparse*-labeling approach, where only a few scale worms per image were annotated. Patches identified from the sparse labeling procedure were used as training patches. The training patches were augmented via rotation, width shift, height shift, shear, zoom, and horizontal flip. As U-Net works on a pixel-level, no empty patches were used for training.

A test-set consisting of 45 images, with equal representation from the three CamHD scenes, was used for network testing. However, this test set was *densely* labeled, so that every visible scale worm in each image was annotated. Note that dense labeling of scale worms in these CamHD images can be taxing, as some scale worms are very difficult to detect, even for skilled human observers. Our full labeled data is available online.[1]

During U-Net data processing, it was observed that scale worms can be easily confused for tube worms and other background objects when viewed from a full scale perspective. However, we noted that scale worms were more easily recognizable when patch size was roughly equivalent to the size of scale worms. Our pipeline, therefore, was trained using small patch sizes, provided the patch size remains large enough to encompass an entire scale worm. Ideally, we believe, patch size will be approximately equivalent to the largest potential scale worm detection size. Upon observation over several annotated masks, we determined that a square having each side equal to 256 pixels to be an appropriate patch size, as it was sufficiently large enough to contain most scale worms.

3.2. Region Proposal

CamHD natively produces HD images (1920×1080 pixels). During image inference, the input images were split into individual patches sized 256×256 px^2, via a sliding window with stride size of 128 px (half a patch size). Such an overlapping stride size was chosen to ensure we capture scale worms which may be split across different patches. U-Net masks were then generated from these individual sliding window patches, before being stitched back together to form a single predicted whole-image-mask. As scale worms may be split across two patches, we utilized a center-crop stitch method. An example of this method is shown in Figure 6, where dotted and solid lines (in red) represent two subsequent patches. The scale worm in this example clearly falls between the individual patches. However, when the two patches are concatenated, the complete scale worm segmentation representation is restored, as shown by the dashed lines (green). This method works because the U-Net model learns to produce the segmentation mask at a pixel-level. The connected components in the resultant whole-image-mask are considered as the proposed regions.

[1] https://www.camhd.science/categories/fauna_segmentation/

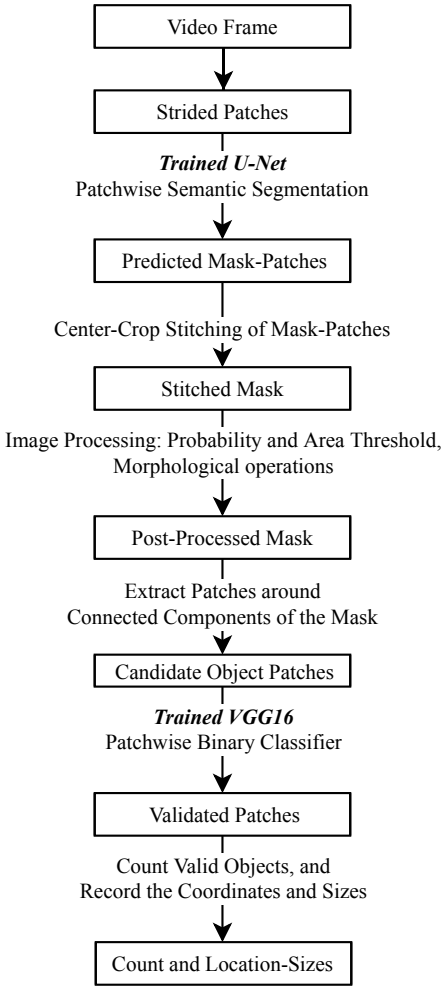

Figure 5: Flowchart depicting the hybrid U-Net + VGG-16 CNN inference pipeline.

3.3. Incorporation of a VGG-16 CNN

The base U-Net model mainly learns color and texture, and also considers the structural information. While the color and texture information is typically useful for net-

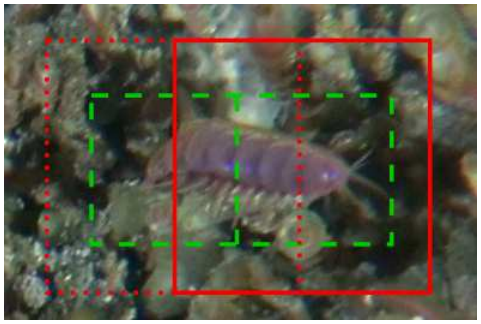

Figure 6: Center-crop stitch method across two subsequent patches.

work performance, the similarities in the appearance between scale worms and the background Mushroom environment led to a high number of false positives from the U-Net output, as shown in Figure 7(b). Probability and area thresholding of this output did help in reducing the false positives, however, they still remained troublingly high (see Figure 7(c)). Our solution was to add a patch-wise binary classification CNN to reduce the false positives from the base U-Net output. We chose a *VGG-16* CNN network, which has been found to be effective on the ImageNet dataset [29]. We additionally add batch-normalization layers to improve convergence and detection accuracy as it helps in limiting potential over-fitting [27, 11]. Our full network architecture, which also includes the VGG-16 CNN network portion, is shown in Figure 5.

3.4. VGG-16 CNN Training

The VGG-16 CNN, operating as a binary classifier, was trained on image patches (still at size 256×256 px^2), outputting a probability score on the likelihood that the proposed region from U-Net should be classified as a scale worm. This network was trained on both valid (i.e. containing a scale worm) and invalid (i.e empty) patches. For valid training images, the network simply used training patches from the U-Net dataset. For invalid patches, we considered two types of data: 1) randomly cropped patches from various video frames, and 2) manually labeled patches in the regions typically occupied by scale worms. Further discussion on these two invalid patch types can be found in section 4.1.

The network was trained to minimize cross-entropy loss using the Adam optimizer with learning rate 10^{-3}. The VGG-16 CNN output was thresholded at 0.7, where scores above that value were classified as scale worms.

3.5. Evaluation Metric

We consider Average Precision at a 50% IoU (AP@[0.5]) as our primary evaluation metric. Scale worms are typically located in neighborhoods having a high tube worm density, and we observed that tube worms and other background objects often occlude a portion of several scale worms. Moreover, when scale worms are close to the tip of a tube worm, it is very hard to manually distinguish the boundary of the scale worm. Therefore, we believe it is reasonable to discount such inaccuracies in our application, and hence we chose an Intersection over Union (IoU) threshold of 50%.

3.6. Ignoring Scale Worms in Training Data

The population of scale worms on Mushroom span a broad range of sizes and colorations; including a significant variation in appearance between individual scale worms. The authors theorize that these individuals come from different species. Furthermore, previous work has found that same-species appearances may vary substantially due to size and sexual dimorphism [12, 9, 32] and age/growth rate differences [7]. An example of the varying scale worm appearance is shown in Figure 8, where one scale worm exhibits a red color, and one exhibits a darker green color. Furthermore, the green-colored scale worms were typically substantially larger than the red ones, occasionally surpassing the 256×256 px^2 patch size. To reduce potential sources of confusion, we ignore these large, green scale worms, which have highly unique appearance, within our initial experiment.

3.7. Inference Pipeline Advantages and Limitations

The presented inference pipeline has several advantages over other methods. First, our approach provides reasonable performance with limited training data. This could be due to the decoupling of the hard instance segmentation problem into two relatively more tractable problems: 1) semantic segmentation, where U-Net is effective with small amounts of training data, and 2) binary classification for validation in order to increase the precision. The limited training data requirement is useful in situations where the collection of large labeled datasets is not feasible.

Second, this pipeline requires only patch-level labeling, enabling it to work effectively with sparse-labeled ground truth video-frames. This is crucial for situations where dense labeling is not feasible, as is the case in the presented Mushroom environment where exhaustive dense tagging of every scale worm in an image is extremely tedious, if not impossible. Furthermore, the patch-wise approach reduces computational requirements while training.

Third, the inference pipeline is easy to understand, debug and maintain, as intermediate outputs are clear (Figures 5 and 9), and each network can be individually tuned and retrained.

However, there are certain limitations to our approach. First, the predicted confidence score corresponding to each instance segmentation is derived completely from the VGG-

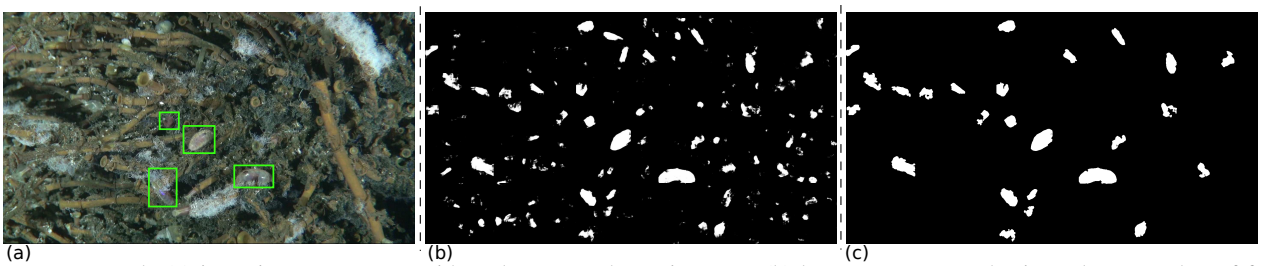

Figure 7: Example (a) input image to U-Net with scale worms shown in green, (b) base U-Net output having a large number of false positives, (c) probability and area thresholded U-net output which has a reduced number of false positives, which is still high.

16 CNN model, and does not quantify the errors from the U-Net model. Second, our experiments were conducted with a relatively small test-set, as large densely-labeled testing datasets were expensive to gather.

4. Results and Discussion

In this section, we discuss the individual performances of our U-Net and VGG-16 models, followed by results from the complete instance segmentation inference pipeline. Finally, we discuss disjoint model training through several experimental observations.

4.1. Training Data Sets

To gauge the amount of training data required for effective U-Net, VGG-16, and complete inference pipeline performance, we split training data into several disjoint sets. Set one (S1) and set two (S2) each contained manually labeled patch-masks (pairs of a 256×256 image patch and the corresponding labeled segmentation mask) from the Mushroom scenes, where these sets contained 261 and 276 patches respectively. Additionally, we identified two types of negative patches (patches that do not contain a scale worm) specifically for VGG-16 network training:

1. *Randomly Cropped Patches* (RCP): negative patches taken randomly from various "empty" Mushroom scenes. These patches were manually inspected to ensure that they contain no (previously unnoticed) scale worms; and

2. Manually labeled *Custom Negative Patches* (CNP): specifically chosen by the authors from Mushroom locations with high false positive rates (as per manual observation), to specify to the model the difference between an actual scale worm and the environment it typically lives in.

   Note that the curation of this dataset was more expensive than the RCP dataset. However, as we will describe in section 4.3.1, we find that these patches significantly improve model performance.

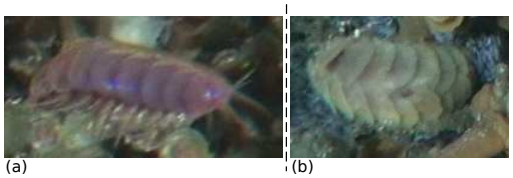

Figure 8: Variations in scale worm appearance. (a) The typical small, 'red' scale worms, and (b) larger 'green' scale worms which were ignored during model training.

Training Data Name	Data
S1	261 custom labeled patch-masks
S2	276 custom labeled patch-masks
RCP	392 randomly cropped negatives
CNP	995 custom negatives

Table 1: Training datasets.

We identified 392 RCP and 995 CNP training patches. These individual training datasets are shown in Table 1.

4.2. Individual Model Performance

Using the datasets described in section 4.1, two different U-Net models were trained: version one (U-V1) with training data from S1, and version two (U-V2) with training data from S1 **and** S2. The converged models produced a mean IoU of 0.9267 and 0.9347 respectively, on the validation set. We observed high recall and low precision from individual U-Net outputs, where many regions with moderately simi-

Version	Train Data	Val. Acc.	Val. m-IoU
U-V1	S1	0.9835	0.9267
VGG-V1	S1+RCP	0.9921	N.A.
U-V2	S1+S2	0.9821	0.9347
VGG-V2	S1+S2+RCP	≈ 1.0	N.A.
VGG-V3	S1+S2+RCP+CNP	0.9947	N.A.

Table 2: U-Net/VGG-16 Inference Pipeline Versions along with the respective individual model performance showing validation accuracy (Val. Acc.) and validation mean-IoU (Val. m-IoU).

lar color and size structures were incorrectly segmented as scale worms.

Similarly, three different VGG-16 CNN models were trained. Version one (VGG-V1) was trained with patches from S1, while versions two (VGG-V2) and three (VGG-V3) were trained with patches from S1 **and** S2. All three of these VGG-16 versions included the RCP patches. Additionally, VGG-V3 was trained by including CNP patches as well. Each of these models had high precision at the 0.7 threshold value, and they all converged to a validation accuracy close to 0.99. The performance and description of each U-Net and VGG-16 CNN model are listed in Table 2. The mean-IoU (m-IoU) criterion, not validation accuracy, is the primary evaluation parameter for the individual U-Net models, whereas validation accuracy is the only evaluation parameter for the individual VGG-16 CNN models. Note that these individual model performance metrics are based on patch-wise data.

In this paper, we do not present AP@[0.5] on the test-set from any individual U-Net *without* a VGG-16 CNN model. This was primarily motivated by empirical evidence from the U-Net output showing a large number of false positives, as shown in Figure 9. Given these high false positive rates, we did not believe a comparison to the complete pipeline was warranted.

4.3. Inference Pipeline Experimental Analysis

The complete inference pipeline (Figure 5) was tested iteratively, with the different U-Net and VGG-16 CNN trained versions from Table 2. The AP@[0.5] was used as the primary model evaluation metric, as discussed in section 3.5.

4.3.1 Full Network Output

The full results of this analysis can be seen Table 3, where the highest average precision occurs at configuration 6, when the U-V2 and VGG-V3 models were utilized. An example inference pipeline input, output, and intermediary network steps, for configuration 6, is shown in Figure 9. In Figure 9(a), a sample frame is strided and fed into U-Net, as described in the flowchart shown in Figure 5. Figure 9(b) shows the raw U-Net output, where false positives are very high. False positives are reduced by applying probability and area thresholding as shown in Figure 9(c), but still remain considerably high. The complete pipeline instance segmentation output is shown in Figure 9(d) after the validation performed by the VGG-16 CNN model, where false positives are completely removed and individual scale worms are identified (green boxes).

Configuration	U-Net + VGG-16 version	AP@[0.5]
1	U-V1 + VGG-V1	0.451
2	U-V1 + VGG-V2	0.556
3	U-V1 + VGG-V3	0.589
4	U-V2 + VGG-V1	0.487
5	U-V2 + VGG-V2	0.563
6	U-V2 + VGG-V3	0.671

Table 3: Inference pipeline AP@[0.5] scores for various pipeline configurations.

4.3.2 False Negatives

While the complete inference pipeline removed the majority of false positives, false negatives (i.e. scale worms not detected by our pipeline) still remain. From section 3.6, the pipeline purposely ignores large, green scale worms, and therefore, we do not consider those missed scale worms to be false negatives (one scale such worm shown in 9(d), lower left of image with gray, dotted bounding box). However, as is shown in Figure 9(d), the network did fail to find some of the smaller scale worms it was trained to identify (shown in red bounding boxes). These failures primarily occurred when the scale worms were in odd poses, e.g., on its side or back. We did not have enough data to properly train for these scale worm orientations. However, we believe that the network will properly identify these scale worms given sufficient training data.

4.4. Impact of Training Data on Model Performance

From Table 3, there is a $\approx 10\%$ increase in AP@[0.5] between configurations 1 and 2, 4 and 5, and 5 and 6 as more data is added. These observations follow intuitive understanding, as increasing data will typically result in better performance.

Interestingly, however, the large AP@[0.5] increase from configuration 5 to configuration 6 occurred without an increase in U-Net model training data. Examining Table 3, the AP@[0.5] appears to, typically, be primarily impacted by the VGG-16 version. For example, the AP@[0.5] of configuration 2 and configuration 5 is approximately equivalent, as the VGG-16 versions were identical (VGG-V2).

This observation is critical, because a primary drawback of a multi-network inference pipeline is the maintenance requirement to retrain multiple networks. If complete pipeline performance can be greatly improved by retraining only one of the models in the pipeline, overall pipeline maintenance becomes much simpler.

Two observed exceptions to the above analyses are:

1. A noticeable AP@[0.5] performance increase between configuration 3 and configuration 6, when the U-Net version **did** change.

2. An almost negligible AP@[0.5] improvement between configuration 2 and configuration 3, despite changing from VGG-V2 to VGG-V3.

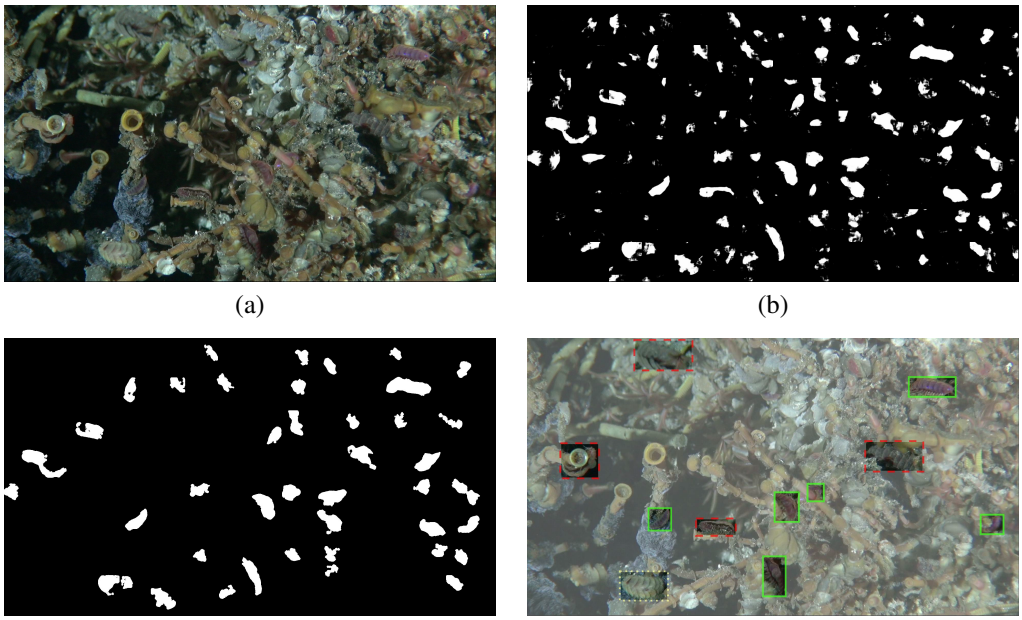

Figure 9: Example intermediate outputs from the inference pipeline (with configuration 6 as shown in Table 3), with (a) example input frame, (b) stitched U-Net output mask, (c) post-processed mask after probability and area threshold, and (d) complete instance segmentation pipeline predictions (solid, green boxes), hand identified false negatives (dashed, red boxes), and one 'large, green' scale worm, not included in pipeline training routine (gray, dotted).

As the only difference between VGG-V2 and VGG-V3 is the inclusion of CNP to the VGG-16 CNN model, the above two observations suggest that a certain quality output from the U-Net model is required before the addition of the CNP training set to the VGG-16 CNN model has a significant impact on the inference pipeline output AP@[0.5]. This result does run counter to our other observations, and suggests that a 'bare minimum' U-Net quality is required before the VGG-16 model dominates the pipeline performance.

5. Conclusion and Future Work

The presented work details an instance segmentation approach for the identification of scale worms from the imagery of a subsea hydrothermal vent. We developed a hybrid U-Net + VGG-16 CNN inference pipeline, designed for instance segmentation of camouflaged objects with limited training data. As the core two networks of our pipeline are disjoint, this method is easy to interpret, debug, and individually train. While fairly simple, this pipeline effectively segments scale worms and is useful for identifying structural information (i.e. size and location) in specific Mushroom scenes. The model was demonstrated to have an AP@[0.5] of 0.671 when trained on our full training dataset.

We furthermore discussed the impact that varying training data had on the complete pipeline performance. We found that the VGG-16 CNN portion of the inference pipeline was more sensitive to the addition of training data, while the U-Net model was typically less improved by increased quantities of training data. This implies that incremental retraining can focus on the VGG-16 CNN model, reducing the quantities of supplemental labeled data required for the U-Net. For dynamic scenes where data addition and retraining is regular (such as the presented scene), reducing model training to only one network simplifies model maintenance.

There are several future work directions. First, the expansion of the application to support multiple-class instance segmentation to capture other genera of macrofauna at Mushroom. Second, transfer-learning based approaches may be impactful, given the limited labeled data. Third, applying the presented pipeline to instance segmentation tasks in other domains to understand if the observations are generalizable. Finally, once a sufficiently robust automatic segmentation algorithm has been achieved, it can be used to extract meaningful spatial and temporal statistics about the scale worm population at Mushroom. These statistics can be correlated with observations of temperature, pressure, tidal cycles, and volcanic activity collected elsewhere on the RCA to increase the study of benthic marine life in these important, yet observationally difficult, subsea environments.

References

[1] Family Polynoidae. https://eol.org/pages/117, 2016.

[2] NSF Ocean Observatories Initiative. https://oceanobservatories.org/, 2018.

[3] NSF Ocean Observatories Initiative Regional Cabled Array. https://oceanobservatories.org/array/cabled-array/, 2018.

[4] Ocean Networks Canada. http://www.oceannetworks.ca/, 2019.

[5] D. Cuvelier, P. Legendre, A. Laes, P.-M. Sarradin, and J. Sarrazin. Rhythms and community dynamics of a hydrothermal tubeworm assemblage at main endeavour field–a multidisciplinary deep-sea observatory approach. *PLoS One*, 9(5):e96924, 2014.

[6] D. Cuvelier, P. Legendre, A. Laës-Huon, P.-M. Sarradin, and J. Sarrazin. Biological and environmental rhythms in (dark) deep-sea hydrothermal ecosystems. *Biogeosciences*, 14(12):2955, 2017.

[7] J. Daly. The maturation and breeding biology of harmothoe imbricata (polychaeta: Polynoidae). *Marine Biology*, 12(1):53–66, 1972.

[8] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation, 2017.

[9] A. G. Glover, E. Goetze, T. G. Dahlgren, and C. R. Smith. Morphology, reproductive biology and genetic structure of the whale-fall and hydrothermal vent specialist, bathykurila guaymasensis pettibone, 1989 (annelida: Polynoidae). *Marine Ecology*, 26(3-4):223–234, 2005.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015.

[12] D. Jollivet, A. Empis, M. Baker, S. Hourdez, T. Comtet, C. Jouin-Toulmond, D. Desbruyres, and P. Tyler. Reproductive biology, sexual dimorphism, and population structure of the deep sea hydrothermal vent scale-worm, branchipolynoe seepensis (polychaeta: Polynoidae). *Journal of the Marine Biological Association of the United Kingdom*, 80:55–68, 2000.

[13] H. Lamba. Understanding Semantic Segmentation with UNET. https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47, 2019.

[14] C. Langmuir, S. Humphris, D. Fornari, C. V. Dover, K. V. Damm, M. Tivey, D. Colodner, J.-L. Charlou, D. Desonie, C. Wilson, Y. Fouquet, G. Klinkhammer, and H. Bougault. Hydrothermal vents near a mantle hot spot: the lucky strike vent field at 37n on the mid-atlantic ridge. *Earth and Planetary Science Letters*, 148(1):69 – 91, 1997.

[15] R. W. Lee, K. Robert, M. Matabos, A. E. Bates, and S. K. Juniper. Temporal and spatial variation in temperature experienced by macrofauna at main endeavour hydrothermal vent field. *Deep Sea Research Part I: Oceanographic Research Papers*, 106:154–166, 2015.

[16] Y. Lelièvre, P. Legendre, M. Matabos, S. Mihály, R. W. Lee, P.-M. Sarradin, C. P. Arango, and J. Sarrazin. Astronomical and atmospheric impacts on deep-sea hydrothermal vent invertebrates. *Proceedings of the Royal Society B: Biological Sciences*, 284(1852):20162123, 2017.

[17] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.

[19] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725, 2018.

[20] Ocean Networks Canada. Observatories. http://www.oceannetworks.ca/observatories/pacific, 2019.

[21] PMEL Earth-Oceans Interactions Program. Axial Seamount. https://www.pmel.noaa.gov/eoi/axial_site.html.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[23] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[24] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):11371149, Jun 2017.

[25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[26] C. Shi, C. Jia, and Z. Chen. Ffdet: a fully convolutional network for coral reef fish detection by layer fusion. In *2018 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2018.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer. Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna. *Scientific data*, 2:150026, 2015.

[29] M. ul Hassan. Vgg16 convolutional network for classification and detection. https://neurohive.io/en/popular-networks/vgg16/, 2018.

[30] K. Wada. labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme, 2016.

[31] W. Xu and S. Matzner. Underwater fish detection using deep learning for water power applications. *arXiv preprint arXiv:1811.01494*, 2018.

[32] Y. Zhang, C. Chen, and J.-W. Qiu. Sexually dimorphic scale worms (annelida: Polynoidae) from hydrothermal vents in the okinawa trough: Two new species and two new sex morphs. *Frontiers in Marine Science*, 5:112, 2018.

[33] Zooniverse. Snapshot Serengeti. https://www.zooniverse.org/projects/zooniverse/snapshot-serengeti.