Enhancing Coral Reef Monitoring Utilizing a Deep Semi-Supervised Learning Approach

Md Modasshir and Ioannis Rekleitis
Computer Science and Engineering, University of South Carolina

***Abstract*— Coral species detection underwater is a challenging problem. There are many cases when even the experts (marine biologists) fail to recognize corals, hence limiting ground truth annotation for training a robust detection system. Identifying coral species is fundamental for enabling the monitoring of coral reefs, a task currently performed by humans, which can be automated with the use of underwater robots. By employing temporal cues using a tracker on a high confidence prediction by a convolutional neural network-based object detector, we augment the collected dataset for the retraining of the object detector. However, using trackers to extract examples also introduces hard or mislabelled samples, which is counterproductive and will deteriorate the performance of the detector. In this work, we show that employing a simple deep neural network to filter out hard or mislabelled samples can help regulate sample extraction. We empirically evaluate our approach in a coral object dataset, collected via an Autonomous Underwater Vehicle (AUV) and human divers, that shows the benefit of incorporating extracted examples obtained from tracking. This work also demonstrates how controlling sample generation by tracking using a simple deep neural network can further improve an object detector.**

I. INTRODUCTION

Coral reefs are an integral part of the marine ecosystem. Coral reefs are also the habitat of numerous marine species [1]. However, coral reefs are rapidly decreasing in area and marine population due to climate change and ocean pollution. Global temperature will increase by $2-4.5^{\circ}$C as per recent predictions from scientists. Due to this dire condition, marine biologists are closely monitoring the coral reefs. Their methods of monitoring coral reefs include scuba divers and autonomous or remotely operated vehicles to capture visual data of the coral reefs. Usually, a transect over a coral reef is surveyed, and afterward, the data are manually annotated according to coral species. Later, experts analyze the annotated data to determine the health and the population diversity of the coral reefs. Such a procedure is tedious and time-consuming. In order to reduce such offline tasks, there have been efforts to automate annotation systems [2]–[4] and analysis procedures [5]. However, these systems mentioned above only speed up a part of the entire coral reef monitoring process. The other tedious part, data collection, and online surveying, still relies entirely on human experts. For a fully autonomous surveying system, we need an online detection guided navigation capability. Modasshir *et al.* [6] utilized coral detection along with the Direct Sparse Odometry (DSO) [7] system to create a 3D semantic mapping in order to improve the data acquisition and analysis system. All these systems, to perform optimally, require an excellent object detection algorithm that can detect coral species under challenging situations.

The authors would like to thank the National Science Foundation for its support (NSF 1513203). The authors would like to acknowledge the help of the College of Engineering and Computing, University of South Carolina.

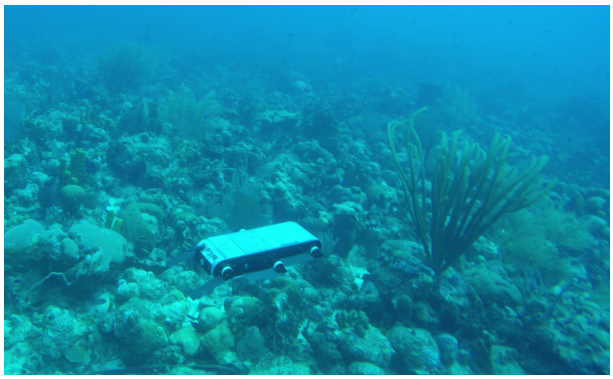

Fig. 1. Aqua2 vehicle navigating over a reef, collecting coral visual data.

Deep learning has revolutionized the object detection systems, resulting in state-of-the-art object detectors in standard vision benchmarks. These object detection algorithms are proven to work well across many domains. In the case of coral detection datasets [2], [4], [8], most of these datasets, being point annotated, are not compatible with object detection algorithms which require bounding box annotation. Hence, we have carefully developed a dataset of Caribbean coral species with bounding box annotations, which were used in our previous work [4], [5]. The annotation of coral species in our dataset was only possible when the coral objects were closer, less blurry, and well-exposed. However, these optimal conditions were not common in the dataset due to hazing, blurring, light variation, and red channel suppression. When we consider monitoring coral reefs via autonomous vehicles, the detection of objects becomes more complicated. Figure 1 presents the Aqua2 Autonomous Underwater Vehicle (AUV) [9] collecting part of our dataset over the coral reefs of Barbados. These complications arise mainly from the nature of the AUV's movement, which introduces motion blur, mostly during rotations. Also, while moving along a transect, the objects near the border of the field of view (FOV) of the AUV are generally further away. Due to poor visibility underwater, objects more than a few meters away are difficult to recognize. Therefore, object detection performs poorly, resulting in a worse analysis of the coral reefs. This issue also presents an opportunity: objects

further ahead and objects near the FOV's left and right corner, are usually equally blurry. Since the transects usually follow a straight line pattern, the further objects straight ahead become more evident over time. Hence, finding a way to augment the training dataset with such objects will help the detection algorithms.

In work by Modasshir *et al.* [6], the semantic mapping system employed both tracking and detection to perform population estimation of coral species while building a 3D semantic map. In the experiments of that work, most of the detections were observed to take place when the objects got closer to the AUV's forward motion. When objects were closer, sometimes detection failed (in a few frames), while tracking succeeded in identifying the objects in-between successful frames. We propose a method to utilize these coral objects carefully, either predicted or tracked, to improve the detection model in a self-training manner. *Self-training* is a strategy where a trained model's predictions are utilized to retrain the model. These tracked and detected corals, henceforth called "soft-labels", are used to augment the training dataset of the detection model. However, inserting all soft-labels into training may also be counterproductive as tracking systems are known to lose track, and the detection model sometimes generates false predictions. Therefore, to constrain the soft-label usage in training, a classification network is used to filter out potentially "harmful" soft-labels.

In this paper, we propose a semi-supervised approach of augmenting an autonomous coral tracking system by utilizing spatio-temporal continuity of the data. Our main contributions are as follows:

- We propose a framework for augmenting the training dataset from coral reef transects. We show how to mine soft-labels by back and forth tracking.
- We construct a constrained loss function to retrain the detection network.
- We demonstrate how a classification network can help filter out "harmful" soft-labels.

The next section reviews related works. Section III introduces the proposed method explaining soft-label generation and incorporation of these labels. Section IV describes the datasets and reports the experimental results to validate the proposed method. Finally, we conclude the paper with future work in section V.

II. Related Works

There are several works on coral classification [3], [4], [8], [10] and detection [5]. In our previous works [4]–[6], [11], we developed a CNN detector and tracker system for coral species. To mine soft-labels for retraining, we utilize the detector developed in [5]. In work by Modasshir *et al.* [5], the RetinaNet [12] was redesigned to train a detector for eight coral species. This work particularly suits our need because the experiments are all performed in the same area of a Caribbean reef with the same species of corals. Hence, we choose the RetinaNet with the settings and the dataset from [5] as our detection model. Among various vision-based trackers [13], the Kernelized Correlation Filters (KCF) [14] performs reasonably online in AUVs [6]. The KCF tracking method is capable of evaluating when an object is out of the field of view while tracking in real-time.

Semi-supervised training has a rich history in computer vision literature [15]–[20]. Among various approaches in semi-supervised methods, the proposed system matches mostly self-training methods [21]–[24]. In a self-training approach, the model is initially trained on fully annotated data then mines for pseudo-labels on weakly-annotated or unlabelled data. The pseudo-labeled data is then used to retrain the model. This process is repeated incrementally [25], [26]. The mining of hard examples has also proven to be useful to create robust object detectors [27]. Recent state-of-the-art object detection algorithms implicitly mine for hard negative examples from unlabelled parts of the images in the training dataset. Rosenberg *et al.* [15] generated pseudo-labeled data by using detections from a pretrained object detector on unlabelled data and then retrained the object detector using these pseudo-labeled data in an incremental procedure. Combining tracking with detection can help generate better and more hard positive examples. In a recent work by Radosavovic *et al.* [28], the authors showed improved performance for the state-of-the-art detectors by incorporating a tremendous amount of pseudo-labeled data. Jin *et al.* [29] used tracking in videos to produce pseudo-labeled data to improve an object detection model. Our work uses forward tracking similar to [29]; however, we also track objects backward to generate relatively harder soft-labeled data. The backward tracked soft-labels are carefully filtered using an outlier rejection network to improve the detection model. In a video sequence, the spatio-temporal cues can be utilized by SLAM systems [30]. In our work, DSO [7] is used to verify tracked objects by feature matching. Aruni *et al.* [31] proposed a soft-label distillation loss function to regulate a retraining procedure by assigning a lower weight to the soft-labels. Down-scaling the effect of the loss of soft-labels allows the network to keep the knowledge of the annotated labels while slowly learning the soft-labeled data and avoids learning incorrect soft-labels.

III. Methodology

Figure 2 presents the overview of the proposed system. Consider frames f_{-m} to f_{n-1}, the system first feeds frame f_0 to the detection model. The bounding box locations with high confidences, box_{locs} from the detection model are then used to initialize the tracking method. Tracking is performed in both direction: backward up to m frames and forward up to $n-1$ frames. At the nth frame, the detection is again performed to retrieve the new bounding box locations, box_{locs}, of the corals. The new bounding box locations are then matched against the tracked bounding boxes, $track_{locs}$. If there is a significant overlap between box_{locs} and $track_{locs}$, then the locations of $track_{locs}$ are modified to reflect the corresponding box_{locs}. Unmatched box_{locs} are used to initialize new trackers instances that search for spatio-temporal cues both in backward and forward direction. For details about the 3D semantic mapping included in the system, please refer to

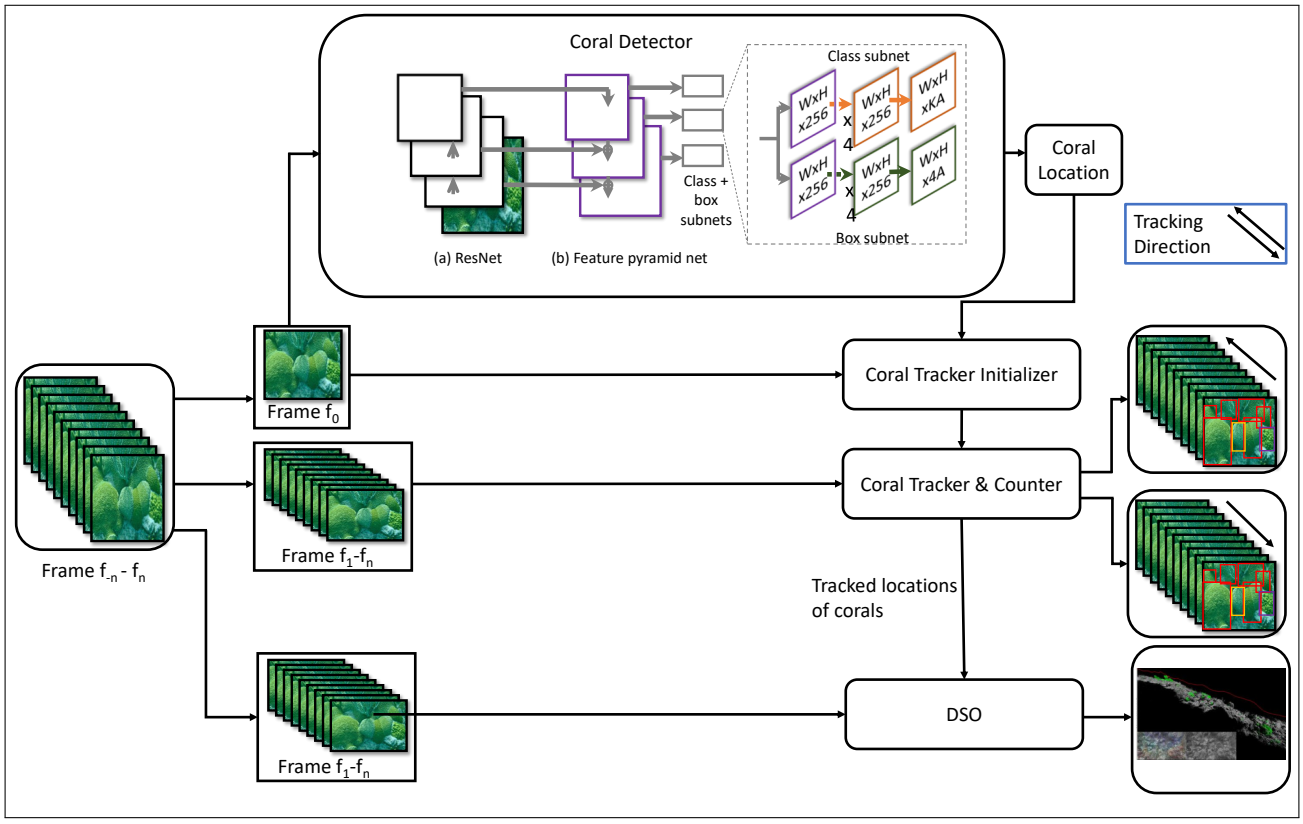

Fig. 2. Overview of the proposed approach.

Modasshir *et al.* [6]. Both box_{locs} and $track_{locs}$ are used to create a soft-labeled dataset $dataset_s$ which we describe in Sec. III-B. Retraining the detection model on both expert-labeled $dataset_e$ and soft-labeled $dataset_s$ is explained in Sec. III-C.

A. Detection and Tracking

The proposed detection model is inspired by RetinaNet [12]. RetinaNet is a single-stage detection model designed on top of a Feature Pyramid Network (FPN) [32]. Utilizing top-down pathway and lateral connections, the FPN enhances a standard CNN with multi-scale features. These multi-scale features enable object detection at different sizes. In our detection model, the FPN builds on top of a ResNet [33] network with a 50 layer variation. There are two sub-networks on top of the FPN for classification and bounding box regression. We redesign the final layer of the classification network to reflect the number of classes in our dataset. The model is optimized using Focal Loss [12] that ensures classes with fewer samples are focused. For a target class, k with estimated probability, $p_k \in [0,1]$, the focal loss is defined as:

$$loss(p_k) = -\alpha(1-p_k)^{\gamma} log(p_k) \quad (1)$$

where $\alpha \in [0,1]$ is the weighting factor and $\gamma \geq 1$ is a focusing hyper-parameter. These hyper-parameters assign lower loss to easily classifiable examples, enabling the model to focus on hard samples. The regression sub-network is trained using smooth L1 loss. Once an object is detected, it is tracked using Kernelized Correlation Filters (KCF) [14].

B. Soft-Label Generation

Soft-labeled samples are generated jointly by the detection model and tracking algorithm. We obtain the predictions box_{loc} on unseen data by the detection model. Then, the predictions with posterior $conf_{score}$ higher than some threshold θ_c, are added to the soft-labeled dataset, $dataset_s$. In our work, we find that setting 0.65 for θ_c works well empirically.

Tracked objects across frames are usually noisy; especially the backward tracked objects. These noisy labels are filtered using two procedures. For forward tracked objects, we obtain the predictions after certain frames, n. We match predictions with confidence scores higher than θ with forward tracked, $track_{loc}$. If the intersection-over-union (IOU) is above a certain threshold, θ_o, and the labels match, we add these samples to $dataset_s$. The IOU is the ratio of the intersection area of box_{loc} and $track_{loc}$ over the union of box_{loc} and $track_{loc}$. Empirically we select 0.7 as θ_o. For backward tracked objects, we observed a very high level of noise. Because these $track_{loc}$ represent the hard instances for the detection model, they are at times detrimental to the detector during training. We back-track up to some m frames and filter using the outlier rejection net III-D before adding to $dataset_s$ except for the very first iteration when we do not have a trained outlier rejection network.

Handling false positive predictions by the detector can

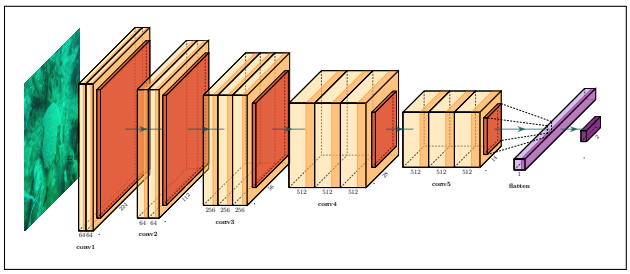

Fig. 3. Outlier Rejection Network

also be used to create soft-labeled samples. Because of the AUV's transect type motions, empirically, we observe correct predictions at a closer range for objects miss-predicted when far away. Therefore, there is inconsistency between the labels of $track_{loc}$ of miss-predicted at frame f_0 and predictions box_{loc} at frame f_n. For such cases, we trust the labels of box_{loc} and relabel the corresponding $track_{loc}$.

C. Training on the Combined Dataset

From the network training perspective, the learning procedure is similar for both expert-labeled $dataset_e$ and soft-labeled $dataset_s$. As the detection model did not recognize the samples in $dataset_s$, it contains harder samples. We intrinsically benefit from the earlier usage of focal loss in our training process as focal loss prioritizes hard examples. However, the deterioration of the performance of the detector on "easy" samples from $dataset_e$ is not desirable. Therefore, we manually put a down-scaling weight ϕ on loss calculated for $dataset_e$ before calculating the total loss $loss_{total}$ on the entire augmented dataset.

$$loss_{total} = loss(p_{ke}) + \phi \cdot loss(p_{ks}) \tag{2}$$

where $loss(p_{ke})$ is the focal loss from Equation 1 from $dataset_e$ and $loss(p_{ks})$ from $dataset_s$.

D. Outlier Rejection Network

It was observed that for backward tracked objects, the samples were mostly unclear, blurry, and less illuminated. Therefore, to filter out samples from $dataset_s$ that are too noisy, a two-way classification network is enough where the network only classifies a sample as "useful" or "harmful" in a coral class agnostic manner. We redesign the VGG16 [34] network to suit this purpose, as shown in Figure 3. The fully connected layers are replaced by an average pooling layer, and the softmax layer is modified to reflect our two classes. The dataset to train this classification network is obtained from the retraining level of our approach. The labels for the samples in $dataset_s$ are obtained by using the training loss on $dataset_s$. For the loss, $loss_{si}$ for a sample s_i is over a threshold θ_s, the label y_i is calculated:

$$y_i = \begin{cases} 1, & \text{if } loss_{si} \geq \theta_s \text{ (discard)} \\ 0, & \text{otherwise (keep).} \end{cases} \tag{3}$$

We choose 0.7 for θ_s based on a few experiments. The model is optimized with cross-entropy loss and a learning rate of 0.001 with a decay of 10^{-7} every 20 epochs.

Dataset	# images	# annotations
GoPro-GT	13523	34510
Aqua-GT	2410	7041
GoPro-Det	20000	36524
GoPro-$Track_{for}$	20000	7484
GoPro-$Track_{back}$	20000	3426
Aqua-Det	1500	3745
Aqua-$Track_{for}$	1500	1421
Aqua-$Track_{back}$	1500	677

TABLE I

DATASET SUMMARY: LISTING THE NUMBER OF IMAGES AND ANNOTATIONS. GOPRO-GT AND AQUA-GT ARE THE EXPERT ANNOTATED DATASET. FOR DETECTION AND TRACKING IN VIDEOS FROM GOPRO AND AQUA, A SUBSET OF THE FIXED NUMBER OF IMAGES WERE USED.

IV. EXPERIMENTAL RESULTS

Experiments were performed using two datasets: GoPro dataset and Aqua2 dataset [5]. We present the quantitative results of detection on these two datasets as well as qualitative detection results on unlabelled data. We also present the coral population density estimation of semantic 3D mapping [6].

A. Dataset Description

GoPro Dataset: The dataset was created with underwater videos of Barbados' coral reef captured by GoPro cameras. The dataset contains annotations for seven different kinds of corals (Brain, Maze, Mustard, Finger, Fire, Star, and Starlet). There are two transects from the GoPro videos: a) 3 minutes 27 seconds over a length of 40.29 meters and b) 10 minutes over a trajectory of 315.39 meters. A scuba diver collected the 3-minute long trajectory, and the 10-minute long transect was captured by using a diver propulsion vehicle (DPV).

Aqua Dataset: The Aqua2 AUV was utilized to capture videos for a region of Barbados' coral reef. The Aqua dataset is also annotated for the same species of corals as the GoPro dataset. The Aqua2 dataset has a trajectory of 1 minute 17 seconds.

Soft-labeled Dataset: High confidence predictions were used to create soft-labels, denoted as *Det* in the dataset description table I. We also collect soft-labels by tracking forward and backward the bounding box locations from the detections. The forward-tracked soft-labels are denoted as $track_{for}$ and the backward-tracked soft-labels as $track_{back}$ in the dataset description table table I. It is worth noting that in forward-tracking, if a tracked object is detected after n frames, the object is labeled as part of the *Det* dataset.

B. Implementation Details

In the experiments, 20% of the GoPro-GT and the Aqua-GT datasets were held out for testing. Firstly, the detector model was trained on the remaining 80% of the GT datasets, referred to as training-GT datasets afterward, following the training procedure described in Modasshir *et al.* [5]. After the training process was completed, the training-GT datasets were then feed-forwarded through the entire system, shown

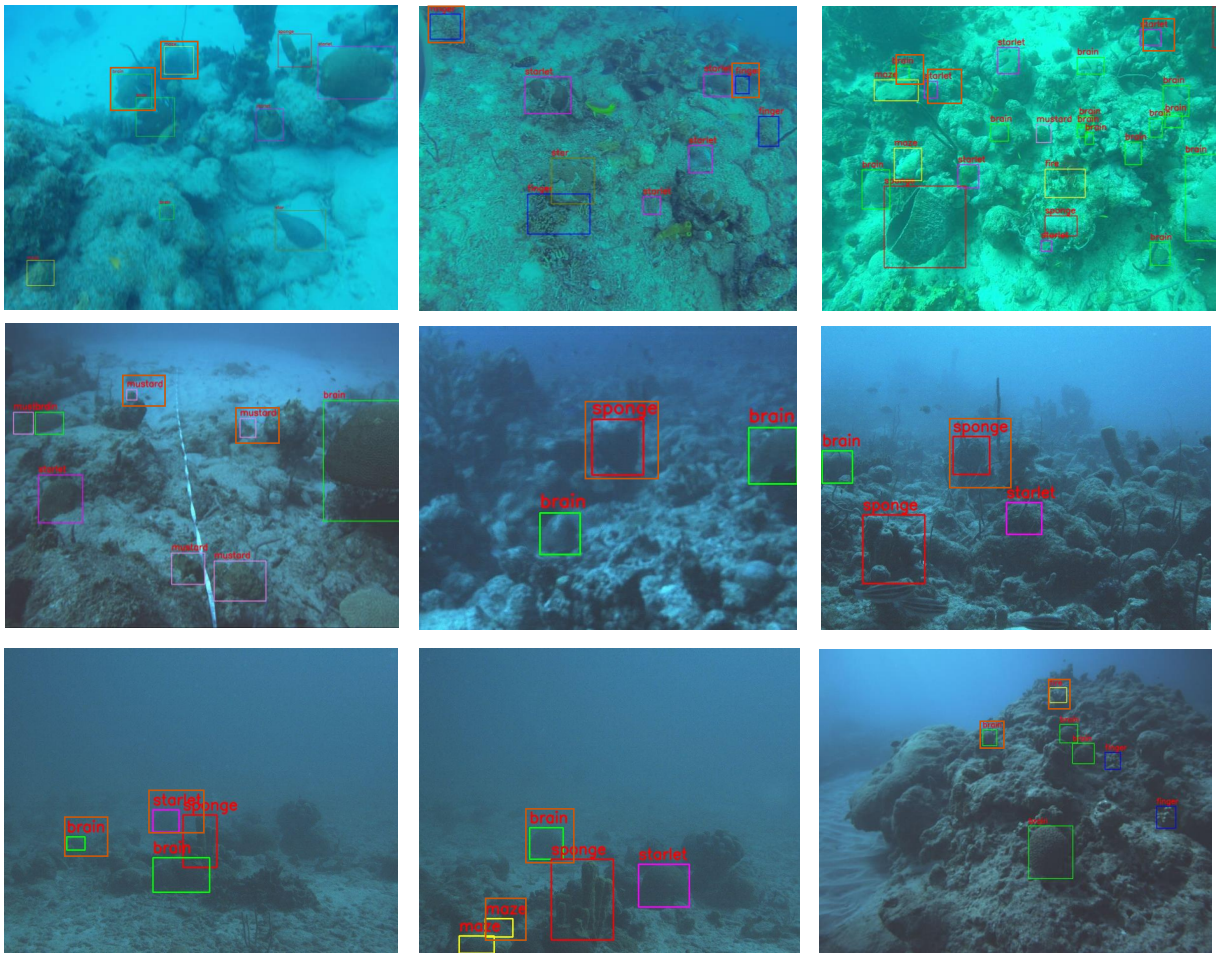

Fig. 4. Detection Results: Corals undetected with the earlier detector of [6], but detected with the proposed approach are marked with an additional orange square. The first row presents images collected by a GoPro camera, while the second two rows present images from an Aqua2 AUV.

in Figure 2 and the confidence scores of detections box_{loc}, and tracked locations in images $track_{loc}$ were recorded. The high confidence predictions (¿70%) were used as soft-labels and incorporated into the training dataset. Finally, the training-GT datasets with expert labels and soft-labels are used to retrain the detection model using the loss function in Equation 2. After retraining is finished, the losses on all samples of training-GT datasets were calculated, and we created another dataset using Equation 3 for the outlier rejection network. This outlier rejection network learns to separate the learnable samples and "very hard" samples and used to remove tough samples from the future retraining processes. The outlier rejection network was used to filter out "very hard" or incorrect samples from the combined training set. The detection model was then trained again, thus beginning the iterative process. About 100 epochs for iterative training are sufficient for the classification network to perform reasonably well.

C. Evaluation of Detection

The results of the evaluation set are presented in table II. On the ground truth data, we were able to achieve average precision (AP) of 24.6 and 13.9 on the GoPro and Aqua datasets correspondingly. We repeat the process of mining for soft-labeled samples, retraining, and outlier rejection. We were able to increase the AP of the GoPro dataset by 16.5 and for the Aqua dataset by 11.4; see Table II.

Figure 4 shows detection results on the evaluation dataset. Empirically, we observe the detections of previously undetected coral objects. Utilizing the soft-labels also improved the detection of coral species with relatively fewer samples, i.e., Fire coral.

Dataset	AP(mean $\pm$ std)
GoPro-GT	24.1 $\pm$ 0.54
Aqua-GT	13.7 $\pm$ 0.21
GoPro-Aug	40.6 $\pm$ 0.92
Aqua-Aug	25.1 $\pm$ 0.67

TABLE II

DETECTION RESULTS ARE PRESENTED IN AVERAGE PRECISION (IOU 0.75) REPORTED AS MEAN AND STANDARD DEVIATION OVER 10 TRAINING ITERATIONS.

D. Evaluation of Counting

Table III shows the comparison of our network against earlier works by Modasshir *et al.* [5], [6]. We observe better detection and quantification of the coral population in the

	Brain	Mustard	Star	Starlet	Maze	Sponge
3min [6]	31/37	0/2	5/7	37/43	11/15	17/17
3min (proposed approach)	35/37	1/2	7/7	41/43	14/15	17/17
10min [6]	90/97	39/47	68/75	161/176	26/28	5/6
10min (proposed approach)	94/97	44/47	77/75	177/176	28/28	6/6

TABLE III

CORAL COUNTING FOR TWO DIFFERENT TRAJECTORIES. CNN-PREDICTION/HUMAN-ANNOTATED. WE PRESENT THE RESULTS BEFORE THE PROPOSED AUGMENTATION; SEE [6], AND THE PROPOSED APPROACH.

transects. In some cases, the counting by CNN based system exceeded human-annotated numbers, i.e., star and starlet coral in 10-minute trajectory. Detection of hard instances by the network resulted in over-count. However, it was difficult to assess whether the network predictions are correct, since revisiting the same coral reef in Barbados was not possible during experimentation. Other than these over-count, the system estimated the coral population fairly accurately.

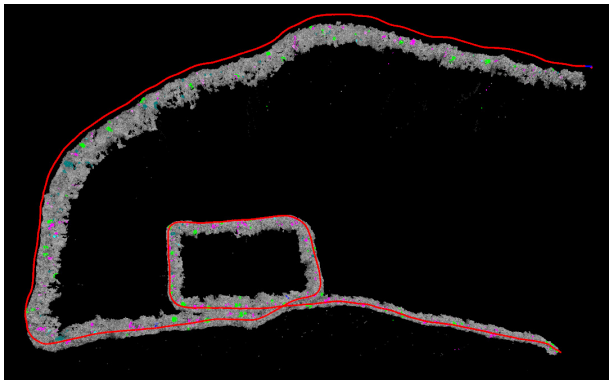

Fig. 5. 3D semantic map of the 10 minute trajectory. Features from different corals are displayed in different colors according to Table IV.

E. Evaluation of Mapping

Figure 5 shows the result of the 3D semantic mapping for the 10-minute trajectory. The reconstruction of coral species was denser and more precise owing to the better bounding box prediction of the improved detector. The color-codes for each coral classes used in the reconstruction are given in Table IV.

Coral:	Brain	Mustard	Star	Starlet	Maze	Sponge
Color:	Green	Purple	Teal	Magenta	Aqua	Blue

TABLE IV

COLOR CODES USED IN SEMANTIC MAPPING FOR DIFFERENT CORALS.

V. CONCLUSIONS

In this paper, we present a novel way to facilitate the coral species annotation from visual data collected during autonomous operation over coral reefs. Extensive data collected using the Aqua2 AUV and a GoPro camera on a Diver Propulsion Vehicle (DPV), see Figure 6, have been used to validate the proposed approach. Furthermore, the collected data have been re-annotated to produce a robust

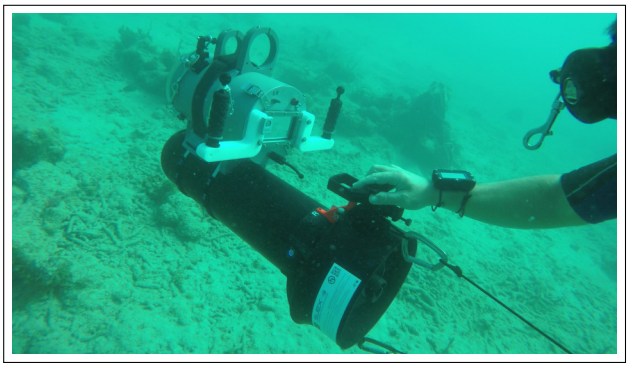

Fig. 6. A custom-made stereo camera suite and a GoPro camera mounted on a DPV during data collection.

dataset of Caribbean corals. By utilizing the spatio-temporal cohesiveness of the data, as collected by an autonomous robot, we demonstrated that object consistency can make the identification process more robust.

Utilizing the proposed approach, autonomous coral reef mapping will be extended to cover longer trajectories and included currently underrepresented species. Future work will integrate the Acoustic, Visual, Inertial state estimation presented by Rahman *et al.* [35] utilizing loop closures for the robust estimation of the observing robot's trajectory in conjunction with the navigation capabilities presented in Xanthidis *et al.* [36] for navigating close to the corals (higher detection rate) while following pre-specified trajectories. Autonomous operations of AUVs [9] will enable the collection of coral population data in a systematic manner. Furthermore, a framework to combine point-annotated coral with box-annotated ones, will provide the marine biologist community with a tool to analyze additional data.

REFERENCES

[1] C. Rogers, G. Garrison, R. Grober, Z. Hillis, and M. F.ie, "Coral reef monitoring manual for the caribbean and western atlantic," *Virgin Islands National Park, 110 p. Ilus.*, 1994.

[2] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1170–1177.

[3] A. Mahmood *et al.*, "Coral classification with hybrid feature representations," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2016, pp. 519–523.

[4] M. Modasshir, A. Quattrini Li, and I. Rekleitis, "MDNet: Multi-Patch Dense Network for Coral Classification," in *OCEANS 2018 MTS/IEEE Charleston*, Oct 2018, pp. 1–6.

[5] M. Modasshir, S. Rahman, O. Youngquist, and I. Rekleitis, "Coral Identification and Counting with an Autonomous Underwater Vehi-

cle," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Kuala Lumpur, Malaysia, Dec. 2018, pp. 524–529.

[6] M. Modasshir, S. Rahman, and I. Rekleitis, "Autonomous 3D Semantic Mapping of Coral Reefs," in *12th Conference on Field and Service Robotics (FSR)*, Tokyo, Japan, Aug. 2019, p. EasyChair Preprint no. 1493.

[7] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, 2018.

[8] O. Beijbom *et al.*, "Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation," *PloS one*, vol. 10, no. 7, p. e0130312, 2015.

[9] J. Sattar, G. Dudek, O. Chiu, I. Rekleitis, P. Giguere, A. Mills, N. Plamondon, C. Prahacs, Y. Girdhar, M. Nahon, and J.-P. Lobos, "Enabling autonomous capabilities in underwater robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, 2008, pp. 3628 – 3634.

[10] T. Manderson, J. Li, N. Dudek, D. Meger, and G. Dudek, "Robotic coral reef health assessment using automated image analysis," *Journal of Field Robotics*, vol. 34, no. 1, pp. 170–187, 2017. [Online]. Available: http://dx.doi.org/10.1002/rob.21698

[11] M. Modasshir, A. Q. Li, and I. Rekleitis, "Deep neural networks: a comparison on different computing platforms," in *Conference on Computer and Robot Vision*, Toronto, ON, Canada, May 2018, pp. 383–389.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[13] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognition*, vol. 76, pp. 323–338, 2018.

[14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[15] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models." *WACV/MOTION*, vol. 2, 2005.

[16] R. Fergus, P. Perona, A. Zisserman, *et al.*, "Object class recognition by unsupervised scale-invariant learning," in *CVPR (2)*, 2003, pp. 264–271.

[17] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow, "Realistic evaluation of semi-supervised learning algorithms," 2018.

[18] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *European conference on computer vision*. Springer, 2000, pp. 18–32.

[19] S. Baluja, "Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data," in *Advances in Neural Information Processing Systems*, 1999, pp. 854–860.

[20] A. Levin, P. A. Viola, and Y. Freund, "Unsupervised improvement of visual detectors using co-training." in *ICCV*, vol. 1, 2003, p. 2.

[28] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He, "Data distillation: Towards omni-supervised learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4119–4128.

[21] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[22] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. Citeseer, 1998, pp. 92–100.

[23] J. Islam, "Towards ai-assisted disease diagnosis: learning deep feature representations for medical image analysis," Ph.D. dissertation, Georgia State University, 2019.

[24] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, vol. 3, 2013, p. 2.

[25] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Cikm*, vol. 5, 2000, p. 3.

[26] F. Muhlenbach, S. Lallich, and D. A. Zighed, "Identifying and handling mislabelled instances," *Journal of Intelligent Information Systems*, vol. 22, no. 1, pp. 89–109, 2004.

[27] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.

[29] S. Jin, A. RoyChowdhury, H. Jiang, A. Singh, A. Prasad, D. Chakraborty, and E. Learned-Miller, "Unsupervised hard example mining from videos for improved object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 307–324.

[30] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017.

[31] A. RoyChowdhury, P. Chakrabarty, A. Singh, S. Jin, H. Jiang, L. Cao, and E. Learned-Miller, "Automatic adaptation of object detectors to new domains using self-training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 780–790.

[32] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection." in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[35] S. Rahman, A. Quattrini Li, and I. Rekleitis, "An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, (IROS ICROS Best Application Paper Award. Finalist), Nov. 2019, pp. 1861–1868.

[36] M. Xanthidis, N. Karapetyan, H. Damron, S. Rahman, J. Johnson, A. O'Connell, J. O'Kane, and I. Rekleitis, "Navigation in the presence of obstacles for an agile autonomous underwater vehicle," in *IEEE International Conference on Robotics and Automation*, 2020, p. accepted.