

One-Shot Informed Robotic Visual Search in the Wild

Karim Koreitem¹, Florian Shkurti¹, Travis Manderson², Wei-Di Chang²,
Juan Camilo Gamboa Higuera², and Gregory Dudek²

Abstract—We consider the task of underwater robot navigation for the purpose of collecting scientifically-relevant video data for environmental monitoring. The majority of field robots that currently perform monitoring tasks in unstructured natural environments navigate via path-tracking a pre-specified sequence of waypoints. Although this navigation method is often necessary, it is limiting because the robot does not have a model of what the scientist deems to be relevant visual observations. Thus, the robot can neither visually search for particular types of objects, nor focus its attention on parts of the scene that might be more relevant than the pre-specified waypoints and viewpoints. In this paper we propose a method that enables informed visual navigation via a learned visual similarity operator that guides the robot’s visual search towards parts of the scene that look like an exemplar image, which is given by the user as a high-level specification for data collection. We propose and evaluate a weakly-supervised video representation learning method that outperforms ImageNet embeddings for similarity tasks in the underwater domain. We also demonstrate the deployment of this similarity operator during informed visual navigation in collaborative environmental monitoring scenarios, in large-scale field trials, where the robot and a human scientist jointly search for relevant visual content.

I. INTRODUCTION

One of the main functions of mobile robots in the context of environmental monitoring is to collect scientifically-relevant data for users who are not experts in robotics, but whose scientific disciplines – oceanography, biology, ecology, geography, among others – increasingly rely on automated data collection by mobile robots carrying scientific sensors [1]. The areas and volumes these field robots are tasked to inspect are too vast to cover exhaustively, so a major challenge in retrieving relevant sensor data is to enable sensing in physical space, in a way that balances *exploration*, to minimize epistemic uncertainty and infer the correct physical model, and *exploitation*, to record data that the user knows they will be interested in. There has been significant progress in addressing the exploration problem in terms of active sensing and informed path planning [2], [3], [4], [5]. Exploitation, however, remains a challenge, because the primary way that users specify and guide the robot’s navigation is by providing a sequence of waypoints that the robot must traverse. This is limiting, however, as it places the

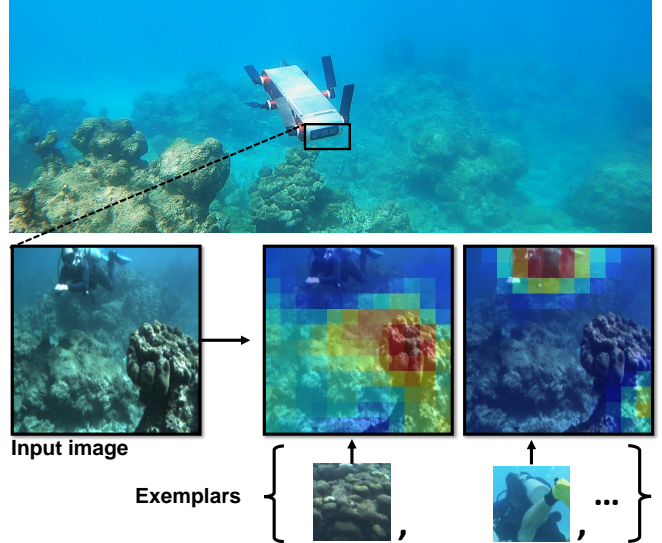


Fig. 1. The output of our visual similarity operator, which is informed by a single exemplar image, given by the user. The exemplar image dictates the behavior of the similarity operator. Note that the exemplars are not objects in the input image; they are objects from the same class, collected from different viewpoints and appearances.

onus of deciding exactly where to look on the user, before deployment, thus allowing for little adaptation.

In this paper, we propose a weakly-supervised method for learning a visual similarity operator, whose output is shown in Fig. 1, which enables informed visual navigation. This similarity operator allows the robot to focus its camera on parts of the scene that the user might deem relevant. Given a set of one or more exemplar images provided by the user, our method finds similar parts of the scene in the robot’s camera view, under diverse viewpoints and appearances. The resulting similarity heatmap is used to guide the robot’s navigation behavior to capture the most relevant parts of the environment, while performing auxiliary navigation tasks, such as visual tracking or obstacle avoidance.

Our method relies on representation learning from videos, and leverages traditional keypoint tracking, which yields patch sequences that locally capture the same part of the scene potentially from multiple viewpoints, as the camera is moving. We then optimize for representations that cluster multiple patch sequences, so that similar sequences are nearby in representation space, while non-similar sequences are pushed apart, via the triplet loss (see Fig. 2).

The main contributions of our paper are threefold: (a) We show that visual similarity operators can be trained to account for multiple viewpoints and appearances, not just

¹Karim Koreitem k.koreitem@mail.utoronto.ca and Florian Shkurti florian@cs.toronto.edu are affiliated with the Department of Computer Science, the Robotics Institute at the University of Toronto, and Vector Institute.

²Travis Manderson, Wei-Di Chang, Juan Camilo Gamboa Higuera, and Gregory Dudek are affiliated with the Center for Intelligent Machines, School of Computer Science, McGill University, in Montreal. {travism, wchang, gamboa, dudek}@cim.mcgill.ca

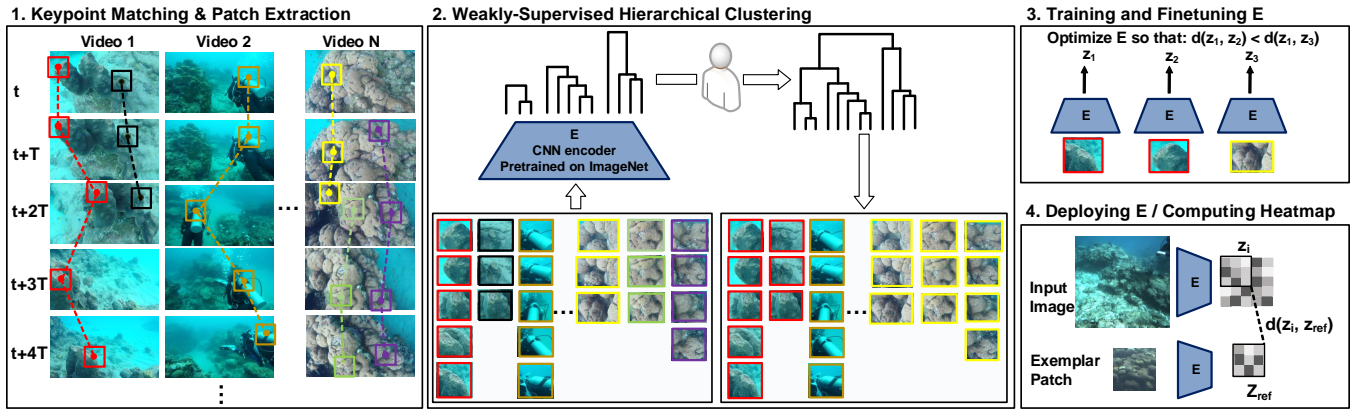


Fig. 2. Overview of our visual similarity training and deployment pipeline.

identify a single object known to exist in the scene. (b) While fully supervised learning techniques for video often require at least a full day of annotations, our method requires about two hours of weak supervision. (c) While most robotic visual search methods are deployed indoors in 2D, or in constrained environments, we demonstrate visual search in 3D, in the open ocean, over hundreds of cubic meters of working volume.

II. RELATED WORK

Image Retrieval and Reverse Image Search: Existing literature on unsupervised representation learning for computer vision is broad. We choose to focus our discussion on similarity learning for visual search and object localization in an image, which has traditionally been a key concern in image retrieval [6] and reverse image search for search engines, and which has been relying on embeddings from convolutional networks [7], [8], [9], [10], [11]. Aside from image retrieval, unsupervised object discovery and localization [12], [13], [14], [15], [16], [17], [18], [19], [20] has also made use of learned visual representations, trained on auxiliary tasks. These tasks typically include ImageNet classification, unsupervised reconstruction-based losses, and egomotion-based prediction for feature extraction [21], [22], [23], [24]. Our method is most similar to [24], however, we use weakly-supervised hierarchical clustering to increase the quality of selecting similar vs dissimilar examples.

Visual Attention Models: Computational models of visual attention [25] in robotics promise to increase the efficiency of visual search and visual exploration by enabling robots to focus their field of view and sensory stream towards parts of the scene that are informative according to some definition [26]. So called *bottom-up* attention methods define an image region as salient if it is significantly different compared to its surroundings or from natural statistics. Some of the basic features for bottom-up attention have included intensity gradient, shading, glossiness, color, motion [26], [27], [28] as well as mutual information [29], [30].

The ability of bottom-up features to determine and predict fixations is not widely accepted [31], especially when a top-down task is specified, such as “find all the people in the

scene”. *Top-down* attention models are task oriented, with knowledge coming externally from a user that is looking for a particular object. Some of the first attention models to have combined these two types of attention are: Wolfe’s Guided Search Model [32] which comprises a set of heuristics for weighing bottom-up feature maps, and the Discriminative Saliency Model [33], which defines top-down cues as the feature maps that minimize the classification error of classes specified by the user. There’s also the Contextual Guidance Model [34], [35] uses the *gist* descriptor [36] that provides a summary of the entire scene to guide the set of possible locations where the desired target might be. Finally, there’s the Selective Tuning Model [37], which relies on a hierarchical pyramid of feature maps, the top of which is biased or determined by a task and the lower levels of which are pruned according to whether they contribute to the winner-takes-all or soft-max processes that are applied from one level of the hierarchy to the next. Notably, it does not only operate in a feedforward fashion.

Visual Search: We want to reward the robot for recording images that are important to the user. Combining bottom-up and top-down attention mechanisms for the purpose of getting the user the data they need is one of our objectives in this work. This has connections to existing active vision and visual search systems for particular objects [38], [39], [40], [41], [42], [43], [44]. Top-down task specification expresses the user’s evolving preferences about what kind of visual content is desired. We treat this bottom-up and top-down visual attention model as a user-tunable *reward function/saliency map* for visual content that guides the robot’s visual search, so that it records more footage of scenes that are deemed important by the user. We contrast *visual search* with *visual exploration* strategies, many of which focus the camera towards parts of the scene that are surprising with respect to a summary of the history of observations [45], [46], [47], [48].

Underwater Navigation: Most existing examples of underwater navigation rely on path and trajectory tracking, given waypoint sequences [49], [50], [51], [2], [3], [1], [4], [5]. Although this navigation strategy is closed-loop with respect to faithfully traversing the waypoints, there

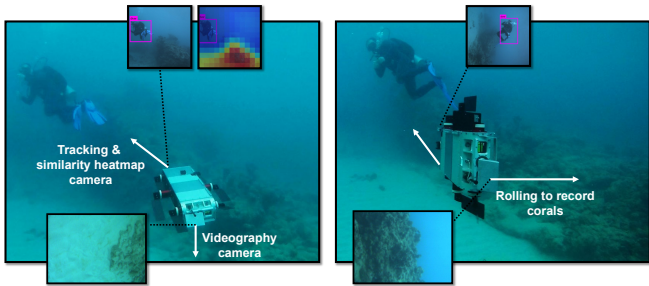


Fig. 3. Overview of the rolling policy employed by the robot, while diver tracking. On the left image, the robot tracks the diver and computes a heatmap based on its front cameras. The videography camera is looking mostly at sand. On the right image, the robot rolls 90 deg, in order to point its downwards-looking videography camera towards the corals, as determined by the heatmap it computed. It maintains diver tracking while doing so.

is no guarantee that useful data will be collected at those waypoints. The robot is unaware of its user’s preferences and objectives for data collection, and it cannot adapt to the scene it encounters.

III. METHODOLOGY

Our goal is to learn a visual similarity operator which will enable more informed visual navigation for underwater robots deployed in the field. In the following sections, we discuss the various subsystems that enable our navigation strategy. In section III-A we present how we leverage large sets of unlabeled underwater videos to automatically extract sequences of patches built from tracked keypoints, which avoids the need for labor-intensive manual semantic annotations. In section III-A.3, we then discuss training a triplet network by ranking triplets built from clusters of the tracked patch sequences to learn a similarity network. Finally, in section III-A.4, we present how the similarity network can be used to build attention heatmaps using exemplars provided by an operator to inform the robot’s navigation. An overview of the system is shown in Fig. 2.

A. Representation Learning from Videos

Due to the low availability of semantically annotated underwater datasets, we opt to rely on unlabeled video datasets. In videos, temporal context can be leveraged as a supervisory signal as object instances will likely appear for more than one frame at a time. Furthermore, consecutive scenes with similar object instances will be available from varied viewpoints and distances, a welcome bonus which will enable the learning of more viewpoint-invariant representations.

1) *Keypoint matching and patch extraction:* In order to obtain semantically similar image patches from videos, we use Oriented FAST and rotated BRIEF (ORB) [52], a fast and robust local appearance-based feature descriptor that allows us to quickly extract similar keypoints across frames. We extract 128 by 128 patches around ORB-extracted keypoints at each frame and track them across the video.

We track our candidate keypoints by brute force matching the descriptors in order to find the best match across consecutive frames and we opt to only keep the top 20 best matches

that satisfy a matching distance threshold. We then save the patches extracted around each tracked keypoint as a tracked patch sequence. This tracked patch sequence will provide the semantic similarity cues that we will leverage when training our triplet network. Depending on the camera’s movements and speed, frame rate adjustments of the video were applied so relevant keypoints can be tracked fast enough as well as to extract patch sequences that had significant variations and avoid very redundant patches (as in a slow moving video).

Certain texture-less areas (such as sand or water) are difficult to track using traditional local descriptors so we instead opt to randomly extract patches from videos mostly containing such areas.

2) *Weakly-supervised hierarchical clustering:* The tracked patch sequences capture very localized notions of similarity. In order to better generalize across environments and viewpoints, we use agglomerative clustering [53] to merge tracked patch sequences that are similar. We use the ResNet 18 [54] model pre-trained on ImageNet and extract the last convolution layer embeddings (conv5) of a randomly sampled patch from each patch sequence and proceed with the clustering. We then build the clusters using the first and last patch in a sequence based on where the randomly sampled patch was clustered. Agglomerative clustering is a bottom-up hierarchical clustering approach where each patch starts as a cluster and patches get merged as we move up the cluster hierarchy. We overcluster our patch sequences, merging only the most similar patch sequences in the process.

Now that the number of clusters is manageable, a human annotator can quickly manually merge clusters. While optional, as shown in section IV, this step provides an additional boost in performance while requiring orders of magnitude less human annotation efforts. After merging, we are left with k clusters forming the set of clusters C .

3) *Training and finetuning:* We build a triplet network [55] using three instances of the ResNet18 [54] architecture neural network as the baseline encoding framework and share parameters across the instances. We use the 18 layers variant of ResNet and sacrifice the accuracy usually gained from using a large variant in favor of runtime speed. We use the ImageNet [56] pre-trained weights as the initialization of the network, discard the fully-connected layers after the conv5 layer. We add an L2 normalization layer and flatten the output. Finally, we finetune the network using a triplet loss [57], [58] which minimizes distance between an anchor and positive patch while maximizing the distance between the anchor and a negative patch.

Using the clusters formed in section III-A we build triplets of patches, (z_A, z_P, z_N) , formed by sampling from the set of clusters C with the following conditions: for $C_k \in C$, $(z_A, z_P) \in C_k$ and $z_N \notin C_k$. The triplet loss we use is defined as follows:

$$L(z_A, z_P, z_N) = \max(0, m + d(z_A, z_P) - d(z_A, z_N)) \quad (1)$$

where d is the cosine similarity between the output flattened descriptors. The margin m enforces a minimum margin

between the $d(z_A, z_P)$ and $d(z_A, z_N)$ and is set to 0.5.

We train a triplet network as opposed to its siamese (pairwise loss) counterpart, which would have relied on pairs of examples, along with a similar/dissimilar label. Using a pairwise contrastive loss implies a need for labels which are contextual, and loses the relative information presented by a triplet. With a pairwise loss, two coral patches of the same species may be considered similar when the dataset includes a large variety in coral species, but dissimilar if the dataset largely consists of the same species and a more refined understanding of similarity is required. On the other hand, using the triplet loss allows us to build triplets where all three coral patches could belong to the same species but are ranked according to how similar their species-specific characteristics are. Given that our training data leverages feature-tracked sequences, we have access to more rich notions of similarity this way.

Implementation Details: We train using the Adam optimizer [59] with a learning rate of 0.0006. During each training step, we perform semihard triplet mining [57] in order to ensure that batches don't exclusively include easy triplets (where $d(A, N)$ is much larger than $d(A, P) + m$). Another limitation is that the number of candidate training triplets grows cubically as the dataset increases. Therefore, we have to sample a subset of the triplets that can be formed across the different clusters. We ensure that a roughly equal distribution of triplets are formed from each cluster. Once trained, the network can be used to extract a $(4, 4, 512)$ descriptor for an RGB patch of dimensions $(128, 128, 3)$.

4) *Deployment and computing heatmaps:* Once learned, the visual similarity network can be deployed in the field to enable more informed robotic navigation by encoding rankings of what a scientist deems important and finding related content. To do this, the operator provides a set of exemplar patches containing visual subjects of interest (in our case, particular species of coral). Given an input image and n exemplars, we build a heatmap H_k for each exemplar z_{ref_k} by doing the following:

- Extract the descriptors of the input image and the exemplar forward passes through the similarity network. Since the input image is larger $(512, 512, 3)$ than the patch image, we get a descriptor of dimensions $(16, 16, 512)$.
- Slide the exemplar patch descriptor over the input image's descriptor, flatten and normalize both descriptors and compute a dot product matrix.
- Upsample the computed similarity matrix to the original image's dimensions to form our heatmap.

An overview of the heatmap construction is presented in part 4 of Fig. 2. Example heatmaps computed from two different exemplars are shown in Fig. 1. We then generate a final weighted heatmap by merging the individual heatmaps along with an optionally-provided ranking of the exemplars, in terms of attention priority. This final heatmap acts as an attention model for the robot to follow as it navigates, prioritizing areas that are most similar to the exemplars provided by the operator.

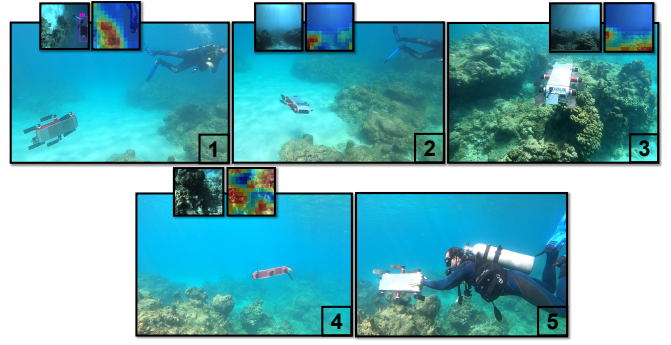


Fig. 4. Overview of the full field experiment run in the Caribbean Sea off the west coast of Barbados. Summary: (1) While diver tracking, the robot finds areas of interest as determined by the heatmap and rolls to point its back camera towards them, for better recording. (2) The robot detaches from the diver and autonomously exploits the area, after detecting a part of the scene that is highly relevant. (3) The robot avoids obstacles based on the visual navigation method in [61] and collects close up footage of the reef. (4) The robot is done exploring when it sees something relevant, about which it decides to alert the diver, and circles around the location of interest until the diver notices (5) The diver manually rolls the robot to 45 deg to signal that he saw the location of interest the robot had identified, and that the robot can continue searching.

B. Informed Visual Search While Diver Tracking

The first way in which we deploy our visual similarity operator on an underwater robot is in the context of visual navigation by tracking a diver. This avoids the need for mapping and localization, and simplifies the navigation process considerably. In this case the robot uses its front cameras to track the diver and compute a heatmap. It uses its back camera, which is downward-looking, as the main videography camera recording in high resolution. This is shown in Fig. 3.

For each incoming front camera frame (at about 10Hz) the heatmap is divided into three regions of equal area. The one that has the highest cumulative score determines the rolling direction of the robot for a fixed number of seconds (in our case, 10s). The rolling direction of the robot is in $[-90, +90]$ deg, and the diver tracking controller [60] is still running. After rolling, the robot returns to its flat orientation to compute the heatmap once again.

C. Informed Visual Search and Autonomous Visual Navigation

The second way in which we deploy our similarity operator on an underwater robot is in the context of autonomous visual navigation that combines obstacle avoidance and relevant-data-seeking behavior. This is shown in Fig. 4.

After the robot identifies a relevant part of the scene during its rolling motion, it decides whether to keep following the diver or detach and start searching on its own. We make this decision based on the heatmap values and coverage in the image. When the robot stops following the diver, it needs to avoid obstacles. We do this via the visual navigation method in [61], based on our existing work. This method learns a vision-based Bayesian Neural Network policy via fully-supervised imitation learning. This policy is a spatial

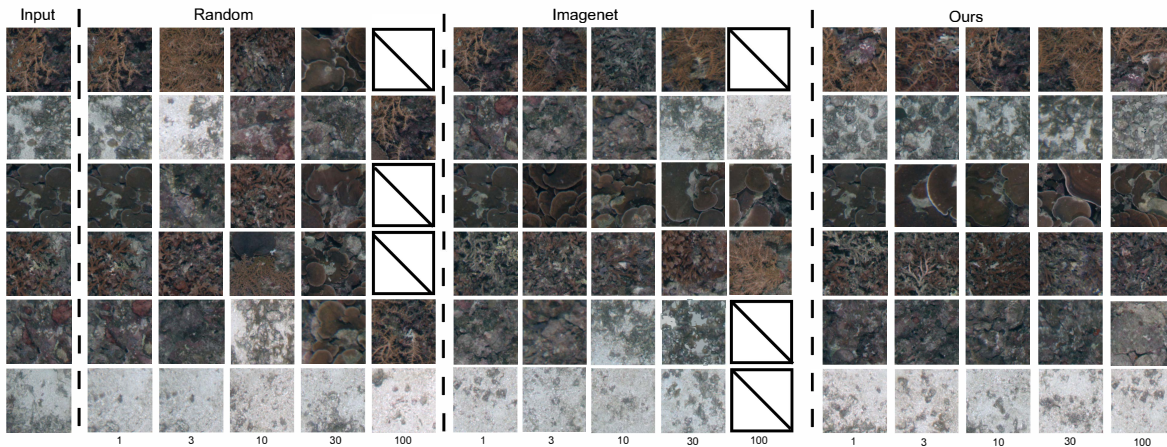


Fig. 5. Example of patches and their ordering in the cluster on *Scott Reef 25*. The exemplar is shown on the left. We then show the 1st, 3rd, 10th, 30th, and 100th retrieved patch. We use our weakly-supervised method and compare with a randomly initialized ResNet18 and pre-trained on ImageNet. Blank striped boxes indicate that the retrieved patches were less than that particular rank.

classifier that outputs the desired direction for pitch and yaw angles of the robot, which are then tracked by the low-level autopilot [50].

The robot keeps searching, using the strategy above, until it sees something highly relevant. In our setup, the robot then needs to alert the diver about the location of interest that it found. Given that we do not have a sufficiently high-power sound source on the robot, we do this by making the robot swim in circles around the location of interest, maintaining depth, until the diver notices. When the diver approaches the robot, it manually rolls the robot to 45 deg, signalling that he has seen the location of interest the robot identified, and that the robot can continue its search, either by tracking the diver or by autonomous navigation.

IV. EVALUATION

We evaluate our system on two underwater datasets as well as in the field. We measure our similarity network’s ability to classify patches when shown exemplars from each class. We also evaluate our system’s ability to generate semantic segmentation by leveraging the heatmaps given exemplar patches from the relevant classes. Finally, we deploy our system on the Aqua robot, an amphibious hexapod robot [62], in an underwater environment and record its ability to collect more relevant images using a camera pointing policy.

A. Underwater Datasets

We perform our evaluations on two underwater datasets for which we train the system separately:

1) *Bellairs Reef dataset*: a collection of 56 videos captured with GoPro and Aqua on-board cameras at different sites in the Caribbean sea off the West coast of Barbados near the McGill Bellairs Research Institute. The videos feature a variety of scenery, viewpoints, and lighting and turbidity conditions. Mainly, they cover footage of sand areas, various dead and live coral and diver-robot activity. The dominant class labels that appear are 1. *Sand*, 2. *Diver*, 3. *Aqua robot*, 4. *Dead coral*, 5. *Finger-like corals (porites, etc)*, and 6. *Spheroid-shape stony corals (brain, starlet, dome,*

etc). We split the dataset into 38 videos used for training and 18 for evaluation. The videos are 5 to 15 minutes long. In general, patch sequences tracked from this dataset span longer timespans as the camera moves slowly, at the pace of an exploring diver. The mean patch sequence length is 4 tracked patches, while the longest sequence spans 30 patches.

2) *Scott Reef 25 dataset*: exclusively top-down stereo camera imagery collected with an AUV at Scott Reef in Western Australia over a 50 by 75 meter full-coverage of the benthos [63], [64]. We use 9831 RGB images captured by the left camera. The images cover areas of dense coral (multiple species), sand and transition areas in between. The dominant class labels that appear in the data are 1. *Sand*, 2. *Mixture*, 3. *Finger-like coral (isopora)*, 4. *Thin birdsnest coral*, 5. *Table coral (acropora)*, and 6. *Dead coral*. We split the dataset temporally into 7061 training images and 2770 test images. Patch sequences tracked from this dataset have 2 patches only on average, mainly due to the high speed of the AUV. The longest sequence spans 23 patches.

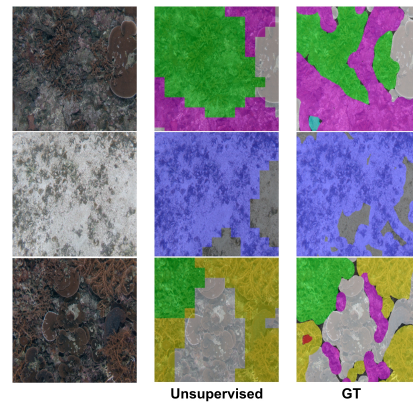


Fig. 6. Example segmentation results on *Scott Reef 25* using our unsupervised model. From left to right: input image, our predicted semantic segmentation, ground truth segmentation (GT).

Model name	Exemplars	Scott Reef 25				Bellairs Reef			
		Acc	P	R	F1	Acc	P	R	F1
CVAE (Weakly-supervised)	1	0.40	0.38	0.40	0.35	0.31	0.37	0.31	0.28
CVAE (Weakly-supervised)	5	0.40	0.39	0.40	0.31	0.31	0.27	0.31	0.25
CVAE (Weakly-supervised)	10	0.43	0.57	0.43	0.32	0.34	0.42	0.34	0.30
IIC (Unsupervised)	NA	0.28	0.14	0.28	0.18	0.33	0.16	0.33	0.21
Randomly Initialized	1	0.24	0.43	0.24	0.13	0.32	0.43	0.32	0.30
Randomly Initialized	5	0.21	0.21	0.21	0.08	0.31	0.25	0.31	0.22
Randomly Initialized	10	0.21	0.15	0.21	0.07	0.33	0.26	0.33	0.26
Pre-trained (Imagenet)	1	0.68	0.74	0.68	0.65	0.67	0.70	0.67	0.68
Pre-trained (Imagenet)	5	0.81	0.83	0.81	0.81	0.73	0.77	0.73	0.71
Pre-trained (Imagenet)	10	0.82	0.86	0.82	0.81	0.71	0.76	0.71	0.70
Unsupervised (ours)	1	0.90	0.91	0.90	0.90	0.62	0.72	0.62	0.62
Unsupervised (ours)	5	0.91	0.92	0.91	0.91	0.69	0.74	0.69	0.68
Unsupervised (ours)	10	0.91	0.92	0.91	0.91	0.66	0.74	0.66	0.66
Weakly-supervised (ours)	1	0.97	0.97	0.97	0.97	0.78	0.80	0.78	0.77
Weakly-supervised (ours)	5	0.97	0.97	0.97	0.97	0.77	0.79	0.77	0.77
Weakly-supervised (ours)	10	0.97	0.97	0.97	0.97	0.77	0.79	0.77	0.77

TABLE I
CLASSIFICATION EVALUATION

B. Representation Learning on Underwater Datasets

For each dataset, we train our network on the automatically clustered and weakly-supervised versions of the dataset as described in section III-A.2. We now evaluate our system on classification and semantic segmentation using standard metrics.

1) *Classification*: We manually curate a subset of 1500 patches from each dataset into classification datasets labelled using the dominant classes listed in section IV-A.1 and section IV-A.2. We then evaluate the system on the classification task by comparing the test set patches with 1, 5, and 10 randomly sampled exemplars of each class and assigning a prediction label to the class with the highest mean similarity. We summarize the classification results in Tab. I. We show that the model demonstrates a boost in performance when finetuned on the clustered patch sequences on both datasets. As expected, by additionally merging clusters manually, we further increase the performance of the system. Note that by comparing against a higher number of exemplars per class, there is no noticeable difference, demonstrating a mostly stable representation of the classes. Only needing a single exemplar is particularly beneficial when deploying on robots with compute limitations. We compare our system against a CVAE [65] and IIC [66] trained on both datasets and outperform them by a significant margin. As a disclaimer, IIC uses an uninitialized ResNet50 architecture while we finetune on an ImageNet pre-trained ResNet18 network, thus benefitting from the pre-trained weights. We used the TensorFlow implementation of IIC¹.

An important note is that the system is able to obtain an ordering of the retrieved patches based on their similarity to the provided exemplar as opposed to simply classifying it. This is especially important when an operator is interested in retrieving images of an object that are not as common as the class center. Example ordered retrieval results are presented in Fig. 5.

2) *Semantic segmentation*: We manually annotate pixel-level semantic segmentation masks for 250 randomly sam-

pled images from each dataset using their respective dominant classes. To build a semantic segmentation prediction, we merge the heatmaps generated for each class by assigning every pixel to the class with the highest similarity. We summarize the results in Table II and show example segmentations in Fig. 6.

Dataset	Model	Mean Acc	Mean IoU	Weighted IoU
Scott Reef 25	Randomly Initialized	0.30	0.03	0.02
	Pre-trained (ImageNet)	0.39	0.28	0.51
	Unsupervised	0.41	0.31	0.57
	Weakly-supervised	0.42	0.29	0.52
Bellairs Reef	Randomly Initialized	0.32	0.20	0.41
	Pre-trained (ImageNet)	0.59	0.40	0.56
	Unsupervised	0.46	0.34	0.54
	Weakly-supervised	0.51	0.36	0.54

TABLE II
SEGMENTATION EVALUATION

C. Robot Field Trials

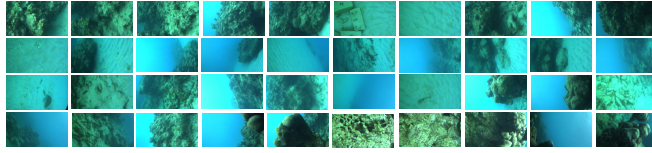
We deploy our system on the Aqua underwater robot [62], in the Caribbean sea, on the west coast of Barbados, over the course of two weeks. The Robot Operating System (ROS) [67] framework is used to handle distributed communications among Aqua’s controllers and sensor suite. We rely on the front-facing RGB camera which runs on a laptop-grade dual-core Intel NUC i3 CPU and an Nvidia GPU (Jetson TX2) [68]. The diver tracker [60], [69] ran on the NUC at 10Hz, while the similarity operator ran on the TX2 at 4Hz. A representative experiment showing the improvement in collected data from executing the informed rolling policy illustrated in Fig. 3 is shown in Fig. 7. The rolling policy informed by the similarity operator records more relevant images than the uninformed (random) rolling policy.

V. CONCLUSIONS

We presented a method to learn a weakly-supervised visual similarity model that enables informed robotic visual navigation in the field given exemplar images provided by

¹Github repo: <https://github.com/nathanin/IIC>

Similarity-based rolling



Random rolling every 10 seconds

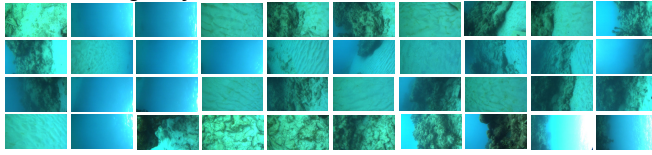


Fig. 7. (Top) Images collected by the back camera of the robot, while doing informed rolling based on the similarity model. 25/40 coral images recorded. (Bottom) Images collected by the same camera at the same place, when making random rolling decisions. 15/40 coral images recorded. The latter sequence contains more irrelevant images (water and sand).

an operator. Our system relies on keypoint tracking across video frames to extract patches from various viewpoints and appearances and leverages hierarchical clustering to build a dataset of similar patch clusters. By asking an operator to weakly supervise the merging of generated image clusters, we circumvent the need for tedious manual frame-by-frame annotations. We then train a triplet network on this dataset, and are able to learn useful representations which enable the computation of an attention heatmap which is used to inform the navigation of a robot underwater. We evaluate our similarity representation's classification and semantic segmentation performance on two underwater datasets and show a boost in retrieval and classification performance compared to simply using pre-trained on ImageNet ResNet18 embeddings. We also successfully deployed our similarity operator on the Aqua underwater robot in large-scale field trials, in which the robot and a diver/scientist collaboratively search for areas of interest and demonstrate a higher retrieval of coral images than uninformed navigation strategies.

ACKNOWLEDGMENT

The authors would like to acknowledge financial support from the Natural Sciences and Engineering Research Council (NSERC) of Canada. The authors would also like to acknowledge the Australian National Research Program (NERP) Marine Biodiversity Hub for the taxonomical labeling and the Australian Centre for Field Robotics for gathering the image data in the *Scott Reef 25* dataset.

REFERENCES

- [1] S. B. Williams, O. Pizarro, D. M. Steinberg, A. Friedman, and M. Bryson, "Reflections on a decade of autonomous underwater vehicles operations for marine survey at the Australian centre for field robotics," *Annual Reviews in Control*, vol. 42, pp. 158 – 165, 2016.
- [2] Y. Girdhar, A. Xu, B. B. Dey, M. Meghiani, F. Shkurti, I. Rekleitis, and G. Dudek, "MARE: Marine Autonomous Robotic Explorer," in *IROS*, San Francisco, USA, sep 2011, pp. 5048 – 5053.
- [3] A. Bender, S. B. Williams, and O. Pizarro, "Autonomous exploration of large-scale benthic environments," in *IEEE International Conference on Robotics and Automation*, May 2013, pp. 390–396.
- [4] O. K. Stephanie Kemna, Sara Kangaslahti and G. S. Sukhatme, "Adaptive sampling: Algorithmic vs. human waypoint selection," in *International Conference on Robotics and Automation*, May 2018.
- [5] A. Stuntz, J. Kelly, and R. N. Smith, "Enabling persistent autonomy for underwater gliders with terrain based navigation," *Frontiers in Robotics and AI - Robotic Control Systems*, vol. 3, 2016.
- [6] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [7] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky, "Neural codes for image retrieval," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, ser. Lecture Notes in Computer Science, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8689. Springer, 2014, pp. 584–599.
- [8] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9910. Springer, 2016, pp. 241–257.
- [9] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *International Journal of Computer Vision*, 10 2016.
- [10] H. Jun, B. Ko, Y. Kim, I. Kim, and J. Kim, "Combination of multiple global descriptors for image retrieval," *arXiv preprint arXiv:1903.10663*, 2019.
- [11] J. Revaud, J. Almazan, R. S. Rezende, and C. R. d. Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- [12] M. Cho, S. Kwak, C. Schmid, and J. Ponce, "Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [13] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural Networks: Tricks of the Trade - Second Edition*, ser. Lecture Notes in Computer Science, G. Montavon, G. B. Orr, and K. Müller, Eds. Springer, 2012, vol. 7700, pp. 561–580.
- [14] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [15] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. USA: IEEE Computer Society, 2015, p. 1422–1430.
- [16] P. Bojanowski and A. Joulin, "Unsupervised learning by predicting noise," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017, pp. 517–526.
- [17] H. V. Vo, F. Bach, M. Cho, K. Han, Y. LeCun, P. Perez, and J. Ponce, "Unsupervised image matching and object discovery as optimization," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [18] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [19] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *European Conference on Computer Vision*, 2018.
- [20] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," *CoRR*, vol. abs/1911.05722, 2019.
- [21] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015, pp. 37–45.
- [22] D. Pathak, R. B. Girshick, P. Dollár, T. Darrell, and B. Hariharan, "Learning features by watching objects move," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 6024–6033.
- [23] C. Xie, Y. Xiang, D. Fox, and Z. Harchaoui, "Object discovery in

- videos as foreground motion clustering,” *CoRR*, vol. abs/1812.02772, 2018.
- [24] X. Wang and A. Gupta, “Unsupervised learning of visual representations using videos,” *CoRR*, vol. abs/1505.00687, 2015.
 - [25] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, Nov 1998.
 - [26] S. Frintrop, E. Rome, and H. I. Christensen, “Computational visual attention systems and their cognitive foundations: A survey,” *ACM Transactions Applied Perception*, vol. 7, no. 1, Jan. 2010.
 - [27] J. K. Tsotsos, *A Computational Perspective on Visual Attention*, 1st ed. The MIT Press, 2011.
 - [28] L. Zhaoping, *Understanding Vision: Theory, Models, and Data*, 1st ed. Oxford Press, 2014.
 - [29] N. D. Bruce and J. K. Tsotsos, “Attention in cognitive systems. theories and systems from an interdisciplinary viewpoint.” Berlin, Heidelberg: Springer-Verlag, 2008, ch. An Information Theoretic Model of Saliency and Visual Search, pp. 171–183.
 - [30] J. Gottlieb, P.-Y. Oudeyer, M. Lopes, and A. Baranes, “Information-seeking, curiosity, and attention: computational and neural mechanisms,” *Trends in Cognitive Sciences*, vol. 17, no. 11, pp. 585 – 593, 2013.
 - [31] G. Underwood, T. Foulsham, E. van Loon, L. Humphreys, and J. Bloyce, “Eye movements during scene inspection: A test of the saliency map hypothesis,” *European Journal of Cognitive Psychology*, vol. 18, no. 3, pp. 321–342, 2006.
 - [32] J. M. Wolfe, “Guided search 2.0 a revised model of visual search,” *Psychonomic Bulletin & Review*, vol. 1, no. 2, pp. 202–238, Jun 1994.
 - [33] D. Gao and N. Vasconcelos, “Discriminant saliency for visual recognition from cluttered scenes,” in *Advances in Neural Information Processing Systems 17*. MIT Press, 2005, pp. 481–488.
 - [34] A. Oliva, A. Torralba, M. S. Castelhana, and J. M. Henderson, “Top-down control of visual attention in object detection,” in *Proceedings 2003 International Conference on Image Processing*, vol. 1, Sept 2003.
 - [35] A. Torralba, M. S. Castelhana, A. Oliva, and J. M. Henderson, “Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search,” *Psychological Review*, vol. 113, p. 2006, 2006.
 - [36] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, May 2001.
 - [37] J. Tsotsos, “Modeling visual attention via selective tuning,” *Artificial Intelligence*, vol. 78, no. 1, pp. 507 – 545, 1995.
 - [38] K. Shubina and J. K. Tsotsos, “Visual search for an object in a 3D environment using a mobile robot,” *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 535–547, may 2010.
 - [39] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, “Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning,” in *International Conference on Robotics and Automation*, 2017.
 - [40] Y. Zhu, D. Gordon, E. Kolve, D. Fox, F.-F. Li, A. Gupta, R. Mottaghi, and A. Farhadi, “Visual Semantic Planning using Deep Successor Representations,” in *ICCV*, 2017.
 - [41] A. Rasouli, P. Lanillos, G. Cheng, and J. K. Tsotsos, “Attention-based active visual search for mobile robots,” *CoRR*, vol. abs/1807.10744, 2018.
 - [42] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, “Coordinated decentralized search for a lost target in a bayesian world,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, Oct 2003, pp. 48–53 vol.1.
 - [43] S. J. Dickinson, H. I. Christensen, J. K. Tsotsos, and G. Olofsson, “Active object recognition integrating attention and viewpoint control,” *Computer Vision and Image Understanding*, vol. 67, no. 3, pp. 239 – 260, 1997.
 - [44] P. Forssen, D. Meger, K. Lai, S. Helmer, J. J. Little, and D. G. Lowe, “Informed visual search: Combining attention and object recognition,” in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 935–942.
 - [45] Y. Girdhar and G. Dudek, *A surprising problem in navigation*, 2011, ch. 11, pp. 228–252.
 - [46] —, “Efficient on-line data summarization using extremum summaries,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 5 2012, pp. 3490–3496.
 - [47] —, “Exploring underwater environments with curiosity,” in *Canadian Conference on Computer and Robot Vision*. IEEE, 5 2014, pp. 104–110.
 - [48] R. Paul, D. Feldman, D. Rus, and P. Newman, “Visual precis generation using coresets,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1304–1311.
 - [49] F. Shkurti, A. Xu, M. Meghiani, J. Gamboa Higuera, Y. Girdhar, P. Giguere, B. Dey, J. Li, A. Kalmbach, C. Prahacs, K. Turgeon, I. Rekleitis, and G. Dudek, “Multi-Domain Monitoring of Marine Environments Using a Heterogeneous Robot Team,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Algarve, Portugal, October 2012, pp. 1747–1753.
 - [50] D. Meger, F. Shkurti, D. C. Poza, P. Giguère, and G. Dudek, “3d trajectory synthesis and control for a legged swimming robot,” in *IEEE International Conference on Robotics and Intelligent Systems*, 2014.
 - [51] M. Meghiani, F. Shkurti, J. C. G. Higuera, A. Kalmbach, D. Whitney, and G. Dudek, “Asymmetric rendezvous search at sea,” in *Proceedings of the 2014 Canadian Conference on Computer and Robot Vision*, 2014, pp. 175–180.
 - [52] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *Proceedings of the 2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.
 - [53] J. H. W. Jr., “Hierarchical grouping to optimize an objective function,” *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.
 - [54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.
 - [55] E. Hoffer and N. Ailon, “Deep metric learning using triplet network,” in *Similarity-Based Pattern Recognition*, A. Feragen, M. Pelillo, and M. Loog, Eds., 2015, pp. 84–92.
 - [56] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F. F. Li, “Imagenet: a large-scale hierarchical image database,” 06 2009, pp. 248–255.
 - [57] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *CVPR*, June 2015, pp. 815–823.
 - [58] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” *CoRR*, vol. abs/1703.07737, 2017.
 - [59] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR*, 2015.
 - [60] F. Shkurti, W. Chang, P. Henderson, M. Islam, J. Gamboa Higuera, J. Li, T. Manderson, A. Xu, G. Dudek, and J. Sattar, “Underwater multi-robot convoying using visual tracking by detection,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, Canada, September 2017, pp. 4189–4196.
 - [61] T. Manderson, J. C. G. Higuera, R. Cheng, and G. Dudek, “Vision-based autonomous underwater swimming in dense coral for combined collision avoidance and target selection,” in *IROS’18*, pp. 1885–1891.
 - [62] J. Sattar, G. Dudek, O. Chiu, I. Rekleitis, P. Giguere, A. Mills, N. Plamondon, C. Prahacs, Y. Girdhar, M. Nahon, et al., “Enabling autonomous capabilities in underwater robotics,” in *IEEE/RSJ IROS*, 2008, pp. 3628–3634.
 - [63] D. M. Steinberg, S. B. Williams, O. Pizarro, and M. V. Jakuba, “Towards autonomous habitat classification using gaussian mixture models,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 4424–4431.
 - [64] M. Bryson, M. Johnson-Roberson, O. Pizarro, and S. Williams, “Automated registration for multi-year robotic surveys of marine benthic habitats,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 3344–3349.
 - [65] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” in *Advances in Neural Information Processing Systems* 28, 2015, pp. 3483–3491.
 - [66] X. Ji, J. F. Henriques, and A. Vedaldi, “Invariant information clustering for unsupervised image classification and segmentation,” ” 2018.
 - [67] O. S. R. Foundation. (2016) Robot operating system.
 - [68] T. Manderson and G. Dudek, “Gpu-assisted learning on an autonomous marine robot for vision-based navigation and image understanding,” in *OCEANS 2018 MTS/IEEE Charleston*, Oct 2018, pp. 1–6.
 - [69] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” *arXiv preprint arXiv:1612.08242*, 2016.