Path to Automating Ocean Health Monitoring

Mak Ahmad,[1] J. Scott Penberthy,[2] Abigail Powell[3]
[1]UC Davis, Davis, CA, USA
[2]Google, Mountain View, CA, USA
[3]Lynker Technologies under contract to NOAA Northwest Fisheries Science Center, Seattle, WA, USA
shahmad@ucdavis.edu, scottpenberthy@google.com, abigail.powell@noaa.gov

Abstract

Marine ecosystems directly and indirectly impact human health, providing benefits such as essential food sources, coastal protection and biomedical compounds. Monitoring changes in marine species is important because impacts such as overfishing, ocean acidification and hypoxic zones can negatively affect both human and ocean health. The US west coast supports a diverse assemblage of deep-sea corals that provide habitats for fish and numerous other invertebrates. Currently, National Oceanic Atmospheric Administration (NOAA) scientists manually track the health of coral species using extractive methods. In this paper, we test the viability of using a machine learning algorithm Convolutional Neural Network (CNN) to automatically classify coral species, using field-collected coral images in collaboration with NOAA. We fine tune the hyperparameters of our model to surpass the human F-score. We also highlight a scalable opportunity to monitor ocean health automatically while preserving corals.

Introduction

Deep-sea corals are a widespread and diverse group that provide complex habitats for many other species (Roberts, Wheeler, and Freiwald 2006). They are distributed in cold water from 4-12°C and can occur as individual colonies or in dense aggregations. Deep-sea coral ecosystems have been shown to be important spawning, nursery and feeding grounds for numerous associated fish species (D'Onghia 2019). Despite their important functional roles, deep-sea corals are vulnerable to damage caused by fishing and changing environmental conditions including rising ocean temperatures and acidity. Monitoring changes in the deep-sea is challenging but important as the health of the oceans is inextricably linked to human health. In addition to providing seafood, the ocean also removes atmospheric CO2 and regulates global temperature, thereby affecting our weather, agriculture and water supply. Consequently, monitoring oceans' health is essential to understanding how environmental changes will impact humans. Improving ocean monitoring technology may contribute to the global effort to maintain a healthy environment for all life. For example, coral imaging contributes to the identification and cataloging of species, enabling scientists to study trends over time in ocean health.

The goal of this study is to design an applied machine learning (ML) application that leverages Google's TensorFlow framework to classify coral species for NOAA. The application uses a fine tuned Inception v3 CNN ML model to classify species from coral images. We analyze test results and various TensorFlow parameters to show the significance of fine tuning the model. This paper will provide a complete ML solution (using coral classification) to be used by field scientists. This solution could be extended to detect coral health automatically. An autonomous underwater drone could capture coral images and send them to a central database where the images could be analyzed by our application. This approach would be a significant step towards the larger goal of monitoring ocean health automatically.

Starting in 2007 Northwest Fisheries Science Center (NWFSC) survey scientists began collecting clippings from corals collected during trawl surveys for genetic analysis and photos of whole specimens for field identification purposes. These collections have resulted in increased knowledge about the distribution and abundance of corals and sea pens along the western coast of the U.S. Although this has resulted in range extensions and discovery of new species, each encounter needs to be classified by either expert taxonomists or by genetic methods. The problem is that this process takes significant time and resources, resulting in delays in providing timely data on the distribution and health of these organisms. We aim to develop an automated classification system which aids the classification process on the vessel.

Related Work

NWFSC scientists are currently doing manual classification on coral samples, as shown in Figure 1, based on reference documents to identify species.

There have been some efforts such as the Video and Image Analytics for Marine Environments (VIAME) to classify fish and scallops however none to classify coral automatically (Dawkins et al. 2017). Coral classification is difficult: many types of corals look alike or are only identified by microscopic characteristics, and some surveys, such as those that utilize bottom trawls, often only retrieve pieces of damaged species, making them difficult to identify.

Figure 1: Sample image of a *Paragorgia arborea* coral

A study based on imagery from shallow coral reefs by Caridade and Marçal (2019) used random forest classification to classify coral reef substrate types. While most research focuses on shallow reefs, deep-sea corals also provide several important ecosystem functions and present unique opportunities to advance ML image processing in low light environments. Ultimately, this study's focus is to provide a framework for a complete end-to-end classification system for use by non-specialists, as opposed to focusing only on image classification schemes themselves.

Methodology

ML Model

We used the Inception v3 model because it has been a top performer in the ImageNet challenge. Major tech companies use neural nets in their core services, such as Facebook's automatic tagging algorithm, Google photo search, and Amazon product recommendations. As humans, we can quickly tell objects apart from images in their environment. A computer, however, has to break images into pixels, where each pixel has 3 RGB values (red, green and blue) for computer displays to reflect the entire visible spectrum. Via pixels, a computer then extracts unique features, starting with edges and curves, and works its way up to paws or whiskers through a series of convolutional layers. This evidence compelled us to choose CNN as our classifier. Moreover, we were encouraged because CNN, inspired by neuroscience, "shares many properties with the visual system of the brain" (Liang and Hu 2015).

Transfer Learning with Inception v3

It is rare to have a large enough dataset to train an entire CNN, so few people train one from scratch. Instead, an existing pre-trained model, Inception v3, can be leveraged and retrained using a custom set of images.

A transfer learning process retrains the Inception v3 model's final layer with the coral images, removing the last fully-connected layer of the inception v3 model (this is where the labeling occurs). It then treats previous layers of the Inception v3 network as a feature extractor for the images. At the last layer, these features are extracted and fed into a linear classifier like linear support vector machines (SVM) or Softmax (used for this study). SVM results are not easy to interpret, but Softmax helps by simply giving the probabilities of each label. For example, for an object detected in an image, SVM classifiers would give the scores [24.2, 0.5, -22.4] whereas Softmax would give [0.9, 0.03, 0.001] which are easier to interpret.

The inception v3 model has 48 layers, based on millions of images, 1,000 categories, 25 million parameters and 5 billion multiply-add operations. These features enable the Inception v3 model to classify an image in fractions of a second. Many other existing models may be leveraged including Microsoft's ResNet, which has 152 layers (104 more than Inception v3), but a worse error rate (3.46% vs 3.57%). We chose inception v3 for its accuracy.

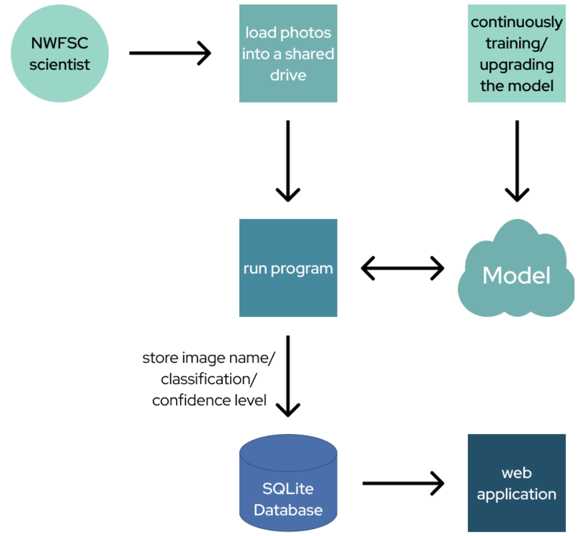

Figure 2: Overall system design

Implementation

We used TensorFlow to re-train the CNN Inception v3 model with custom coral images. This occurs by taking the layer before the final output layer and training it with custom photos. With the model generated, a Python application was developed to classify NWFSC scientists' uploaded photos, move them to separate folders labeled by species name, and to load a database. Another web application has been created that will output the results of the database in a spreadsheet for scientific analysis as shown in Figure 2.

Prepare and Clean Data

We received thousands of images through NWFSC's West Coast Groundfish Bottom Trawl Survey which is a fish-centric survey (Keller, Wallace, and Methot 2017). The bottom trawl net used collects fish and many invertebrates including a variety of coral species. We worked closely with scientists to ensure the barcode and label (species name) were entered accurately into a barcode to species mapping spreadsheet. Using the spreadsheet, we developed a Python

Figure 3: Coral image cropping

Species	**Files**	**Files cleaned**
Acanthoptilum gracile	1,800	123
Anthomastus ritteri	810	32
Anthoptilum grandiflorum	3,180	104
Antipatharia	290	16
Bathypathes sp	450	23
Chrysopathes sp	550	12
Deepsea halipteris	1,830	49
Distichoptilum gracile	250	12
Funiculina quadrangularis	330	19
Gorgonacea	360	3
Halipteris sp	1,910	122
Parastenella ramosa	200	20
Pennatulacea	500	14
Plumarella sp	410	41
Ptilosarcus gurneyi	1,220	36
Stachyptilum superbu	200	19
Swiftia simplex	310	16
Swiftia sp	560	13
Umbellula sp	850	13
Total	16,010	687

Table 1: Coral images selected for modeling

program[1] to parse through the images (labeled by barcode), and move the images into folders (labeled by species). Lastly, we revised the images to species labeling, correcting hundreds of misclassifications due to human error.

A clean training data set is required for a quality machine learning model, so we analyzed each image and cropped it (see Figure 3) to focus on the coral. Some images included multiple corals, while others included hands of scientists or were dark, small or blurry. Out of 16,010 images, 687 were cleaned.

As seen on Table 1, out of 47 total species, 19 were selected for modeling using accuracy and image quality criteria. These 19 species were selected because there were 20+ images of each. Without sufficient training data, the model will be ineffective.

[1]github.com/makahmad/coral-ml/blob/master/parse.py

		Predicted species	
		Species X	Other species
Actual species	Species X	tp	fn
	Other species	fp	tn

Table 2: Evaluation metrics in classification

Creating the Model

The next step is to create the model using the training set, which involves replacing the final layer of Inception v3 with the feature extractor for coral. Retraining the Inception v3 model takes about 30 minutes on a laptop, versus creating a model from scratch, which would take weeks on a powerful computer.

Many techniques may be used to fine tune the model to improve success metrics and thus predictions. Beyond manually cropping images, training steps and learning rate parameters may be fine tuned. Each learning step takes ten random images from the training set and uses the final layer to get predictions. By comparing the predictions with actual labels (species names), the final layer's weights are updated via back-propagation. The learning rate controls the number of updates to the final layer during training. Generally (but not always), the smaller the learning rate, the longer the training time and more precise the model.

Classification Program

With the model created, we created a Python application[2] that classifies unlabeled images and pushes the results into an SQLite database. A benefit of this model is the "ranked" series of species predictions for each image. For example, an image may predict the genus Halipteris with 80% chance. The same image may also predict which Halipteris species with likelihood of 10% chance. These data are saved for each image, beneficial to meeting our goal of creating a coral species recommendation engine for scientists.

Web Application

The final component of the system is a web application[3] installed on NOAA servers to present the results of the predictions to the scientists in a practical manner. The application uses a lightweight Python framework called Weppy and it reads from the SQLite database and returns the results to a table created by the JavaScript framework Datatables. Users will see data on a browser, similar to a spreadsheet. Components are open source, free to use without license.

Results and Error Tuning

For a given species of coral X, a true positive (tp) result occurs when the classifier examines a coral sample and correctly identifies the species as X. If the classifier identifies the sample as species X when the sample is of a different species, then that is a false positive (fp) result.

If the classifier identifies the sample as any species other than X when the sample was of species X, then that is a

[2]github.com/makahmad/coral-ml/blob/master/cnn_classify.py
[3]github.com/makahmad/coral-ml/blob/master/coral_app.py

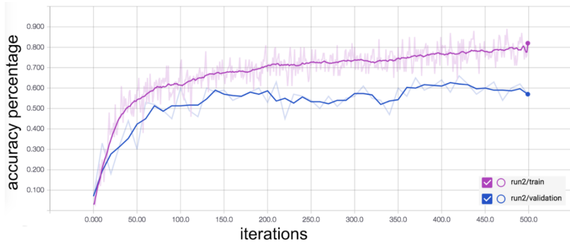

Figure 4: Accuracy graph of model with 500 training steps learning rate of 0.01

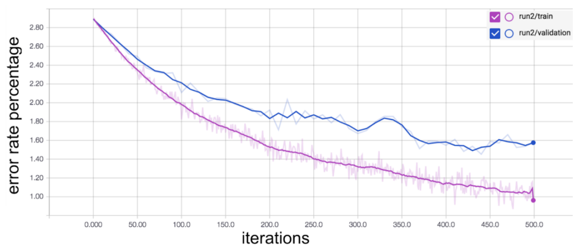

Figure 5: Cross entropy graph of model with 500 training steps learning rate of 0.01

false negative (fn). If the classifier identifies the sample as any species other than X when the sample was not of species X, then that is a true negative (tn). Table 2 summarizes these definitions.

F-score is a method to measure accuracy. It is the weighted average of precision (p) and recall (r), where precision is "the fraction of all positive predictions that are true positives while recall is the fraction of all actual positives that are predicted positive" (Lipton, Elkan, and Narayanaswamy 2014). Precision and recall can be expressed as:

$$p = \frac{tp}{tp + fp} \tag{1}$$

$$r = \frac{tp}{tp + fn} \tag{2}$$

while the F-score (F) can be expressed as:

$$F = 2 * \frac{r * p}{r + p} \tag{3}$$

When manually classifying corals, the F-Score is 70%, obtained from NWFSC scientists. A goal of this paper is to surpass the human F-Score. Numerous tests were conducted; the top three are documented below.

Test Model 0 [prior to data clean up]

Prior to doing any training data cleanup (cropping, deleting blurred images, validating training data), the accuracy and cross entropy (loss) as shown are worse than our test runs with data cleanup built in (see Figures 4 and 5). Models were trained on 80% of our images and tested on the remaining 20%.

The average final F-Score for all species is 69.4%. This was close to our project goal of 70%, but we knew it could be improved.

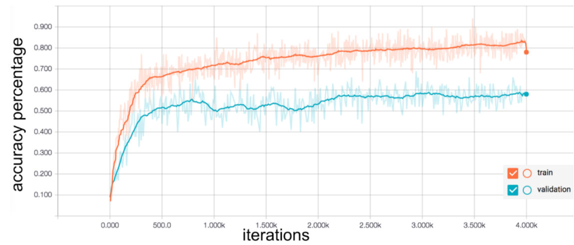

Figure 6: Accuracy graph of model with 4000 training steps learning rate of 0.01

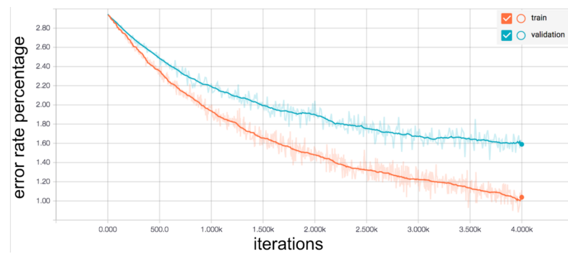

Figure 7: Cross entropy graph of model with 4000 training steps learning rate of 0.01

Test Model 1

In this model, we used 4,000 training steps with a learning rate of 0.01. The validation and training accuracy exhibited (see Figure 6) overfitting (though training data accuracy was higher, and the validation data accuracy curve was flatter).

The cross entropy curves had a downward (as expected) trend (see Figure 7); however, the validation loss could still be improved.

The average F-Score, as seen on Figure 8, for all species is 84%; however, by tweaking more parameters it can be improved. The F-score ranged from 100% to a low of 40% for *Pennatulacea*.

Test Model 2

In this model, we used 8,000 training steps with a learning rate of 0.001. Similar to test model 1, the validation and training accuracy curves exhibited overfitting while the cross entropy curves had a downward trend with room for improvement for validation loss.

The average F-Score for all species was 89%; however, by tweaking parameters, it could still be improved. The F-score ranged from 100% to a low of 30% (again due to *Pennatulacea*).

Test Model 3

In this model, we again used 8,000 training steps with a learning rate of 0.001. We did well in our previous models, however the order *Pennatulacea* skewed our model because we believe the thin shape of these corals (see Figure 9) caused them to blend with the background. Additionally, since *Pennatulacea* contains species that are labeled in other folders, it likely created conflicts preventing accurate prediction. These folders, created from survey data, were set at

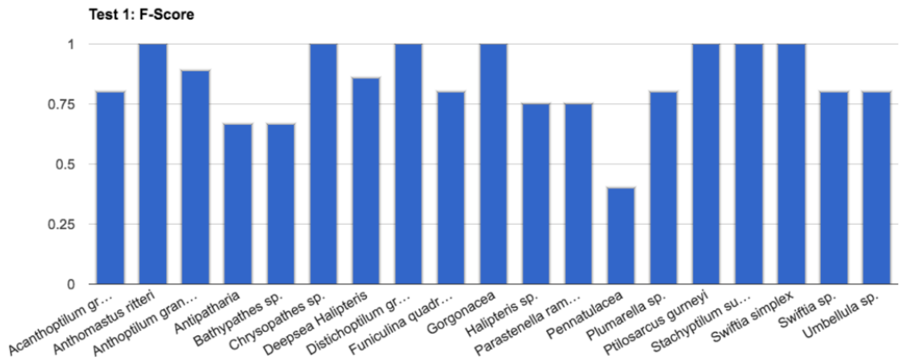

Figure 8: F-score chart of model with 4,000 training steps learning rate of 0.01

Figure 9: Image of *Anthoptilum grandiflorum*

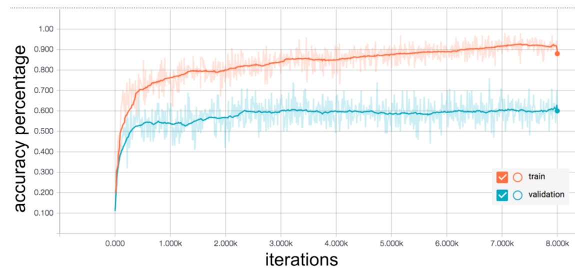

Figure 10: Accuracy graph of model with 8,000 training steps learning rate of 0.001 without *Pennatulacea*

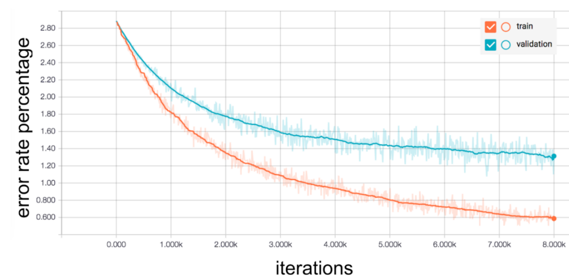

Figure 11: Cross entropy graph of model with 8,000 training steps learning rate of 0.001 without *Pennatulacea*

varied taxonomic levels of accuracy, due to the difficulty of identifying these corals in the field. If we wanted to include *Pennatulacea*, an interactive learning process like multi-task learning can be utilized. We would take an even more refined approach where we predict pixels, which is common in image segmentation.

The validation and training accuracy curves exhibit that there is still overfitting; however, validation accuracy is better than the previous two tests (see Figure 10). The cross entropy has a downward trend (see Figure 11), as expected, with a much improved final loss of only 1.3 (compared to 1.4 and 1.6 in previous tests).

The average final F-Score (see Figure 12) for all species is 96%, greatly exceeding this project's goal of 70% (human achieved F-Score). The F-score ranged from 100% to 65%. When applied to images of corals in the field, the program will provide the score and probability of the closest matching species in the model. The highest scores are obtained when certain species are excluded, when those species are included scores are below 50-60%. Preventing overfitting should also be considered, using dropout and early stopping to observe any improvement in scores.

Limitations and Challenges

There were many challenges in this study. Timeline pressures reduced the amount of time spent organizing, labeling, and cropping thousands of training images. Some images were misclassified, others blurry, some species did not present a sufficient number of images to create a model. Other species were subject to taxonomic changes over the course of collections 2007-2017. Moreover, we were limited by the images from the back decks of 70 foot fishing vessels, which had major inconsistencies with the photography environment (background/lighting/focus/size).

Other challenges included iterations in fine tuning model parameters. In addition to the TensorFlow learning rate and training steps, the following parameters were tuned to reach optimal F-score:

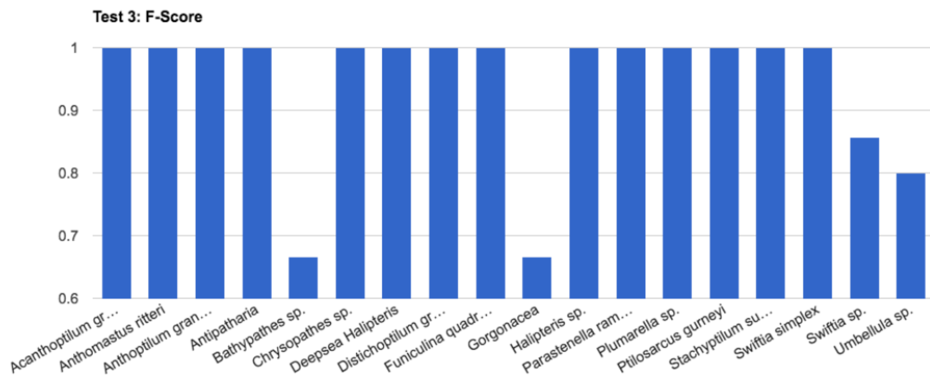

Figure 12: F-score chart of model with 8,000 training steps learning rate of 0.001 without *Pennatulacea*

- **Testing Percentage**: what percentage of images to use as a test set
- **Validation Percentage**: what percentage of images to use as a validation set
- **Test Batch Size**: how many images to test at a time

Other parameters not utilized in this project:

- **Flip Left Right**: whether to randomly flip half of the training images horizontally
- **Random Crop**: a percentage determining how much of a margin to randomly crop off the training images
- **Random Scale**: a percentage determining how much to randomly scale up the size of the training images
- **Random Brightness**: a percentage determining how much to randomly multiply the training image input pixels up or down by
- **Eval Step Interval**: how often to evaluate the training results

From the above list of non-utilized parameters, it would be worthwhile experimenting with random crop, scale, and brightness to reach near 100% accuracy. These parameters might also be helpful when expanding to additional species.

Automation Opportunity

This paper proves that we can leverage image classification techniques to accurately identify coral species purely from images. We are currently doing further research on images collected via remotely operated vehicles (ROVs) to automate gathering of high quality images of live coral, eliminating the coral research scientists' dilemma of analysis vs destruction. We are also exploring the application of these methods on imagery collected by commercial autonomous underwater vehicles (AUVs) which do not require a tether and a support ship. Preliminary results, using imagery collected with underwater cameras, show a 12% deterioration in our model's accuracy. Our hypothesis is that improvement in underwater imaging will drastically improve our model's accuracy. Once we tune our model to the results of the ROVs, we can move on to AUVs. Although there are some challenges to the widespread use of underwater vehicles to carry out surveys, they have become a game-changer (Petillot et al. 2019).

Furthermore, we are extending our species classifier to become a multi-label classifier that can identify species and, more importantly, the health of the coral. The model would have to be trained using healthy vs unhealthy (such as corals with patches of dead tissue) coral images to extend our classifier. Providing this data to NOAA scientists and other global scientists in real time will allow for swift decision making, improving ocean health.

Conclusion

Transfer learning through ImageNet Inception v3 can be used to identify species of coral. This solution can be extended to a multi-label classifier to detect coral health. Further study using Inception v4 may yield a more efficient and accurate classifier. Then, using ROVs and AUVs, we can create an automatic pipeline of images globally, captured and stored in a central database. Lastly, we could leverage visualization tools (Tableau, Looker, etc.) to share insights from this central database, allowing scientists to make critical environmental decisions swiftly. Our proposed approach of capturing images automatically with ROVs and AUVs, storing them centrally, processing them through our multi-label (species and health) classifier, and presenting them to scientists could be a significant step on the path to the automation of ocean health monitoring.

Acknowledgments

The authors thank John Buchanan and Curt E. Whitmire for their coral/trawl survey insights and help analyzing numerous coral images.

References

Caridade, C. M.; and Marçal, A. R. 2019. Automatic Classification of Coral Images using Color and Textures. In *CLEF (Working Notes)*.

Dawkins, M.; Sherrill, L.; Fieldhouse, K.; Hoogs, A.; Richards, B.; Zhang, D.; Prasad, L.; Williams, K.; Lauffenburger, N.; and Wang, G. 2017. An open-source platform for underwater image and video analytics. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 898–906. IEEE.

D'Onghia, G. 2019. 30 Cold-Water Corals as Shelter, Feeding and Life-History Critical Habitats for Fish Species: Ecological Interactions and Fishing Impact. In *Mediterranean Cold-Water Corals: Past, Present and Future*, 335–356. Springer.

Keller, A. A.; Wallace, J. R.; and Methot, R. D. 2017. The Northwest Fisheries Science Center's West Coast Groundfish Bottom Trawl Survey: history, design, and description. *NOAA technical memorandum NMFS-NWFSC ; 136* .

Liang, M.; and Hu, X. 2015. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3367–3375.

Lipton, Z. C.; Elkan, C.; and Narayanaswamy, B. 2014. Thresholding classifiers to maximize f1 score. *arXiv preprint ArXiv:1402.1892* 14.

Petillot, Y. R.; Antonelli, G.; Casalino, G.; and Ferreira, F. 2019. Underwater Robots: From Remotely Operated Vehicles to Intervention-Autonomous Underwater Vehicles. *IEEE Robotics & Automation Magazine* 26(2): 94–101.

Roberts, J. M.; Wheeler, A. J.; and Freiwald, A. 2006. Reefs of the deep: the biology and geology of cold-water coral ecosystems. *Science* 312(5773): 543–547.