CS230 Mid-project report

Elisa Boles – Project partner David Boles (CS undergraduate at Brown University)

Deep learning for coral reef classification from drone imagery

Abstract: Coral reefs are under threat worldwide due to combined stressors of thermal bleaching, ocean acidification, and local pollution and sedimentation. High resolution drone imaging of reefs could provide invaluable information for coral reef practitioners to be able to make decisions efficiently and effectively about management strategies as situations on the reef change. Here, we develop a model based on the U-Net image segmentation approach to identify coral in drone photomosaics from Lighthouse Reef, Palau. The model trained quickly and reached high accuracy (94%) and recall (84%), suggesting that this approach could be readily applied to reefs worldwide.

1 Introduction

Climate change is causing coastal waters to heat up and acidify, threatening coral reefs that support large ecosystems and protect coastlines. Traditionally these reefs have been monitored by manually conducting transects, counting individuals of each species along a line. This technique is labor intensive and thus hard to scale. Advances in imaging platforms, including drones and satellites, as well as computer vision techniques show great promise to reduce the cost and increase the scope of reef monitoring programs. This may be vital to correctly evaluate the impact of different reef management strategies and identify where new ones are needed.

2 Related work

As far as we found, it appears that machine learning applications to coral reef segmentation have thus far focused on high resolution underwater imagery or satellite remote sensing, with no papers yet published on segmentation from drone imagery. Gonzalez-Rivero et al. (2020) investigated fine-tuning a VGG-D 16 network pre-trained on ImageNet for underwater images. Other notable work includes the CoralNet model data repository and collaboration platform run by UCSD (Williams et al., 2019), as well as the Australian Institute of Marine Science's ReefCloud project (https://reefcloud.ai/). These tools have proved valuable to coral ecologists as they allow for easy upload and identification of imagery from the field, and facilitated data sharing and collaboration across institutes.

Two groups have also worked on coral identification from remote sensing data. Remote sensing products are promising as they cover the entire globe . However, other challenges exist for effective use of these large-scale products compared to underwater imagery, namely their lower spatial resolution (> 1 m) and the challenges of surface water reflectance, cloud shading, and wave distortions. Li et al. (2020) of NASA's NeMO-Net group developed a model to segment satellite imagery into a few basic classes (land, sand, coral, etc), which used labels generated from co-located underwater imagery.

3 Dataset and features

The goal of our project is to develop a simple tool to segment images of coral reefs taken by drones. We believe that drone imagery can fill an important gap between underwater imaging and that it is easier to collect large swaths of data than underwater imagery, but higher resolution than satellites. Our dataset comes from a reef in Palau named "Lighthouse Reef," where I performed fieldwork last summer for my Ph.D. The thousands of images taken with the drone were previously collated into one large mosaic covering the entire reef (see Figure 1), which spans about 1 km2 and has a resolution of ~3 inches. The hope is to be able to collect more drone imagery this coming summer and use image segmentation to evaluate how the coral cover has changed over the course of the year.

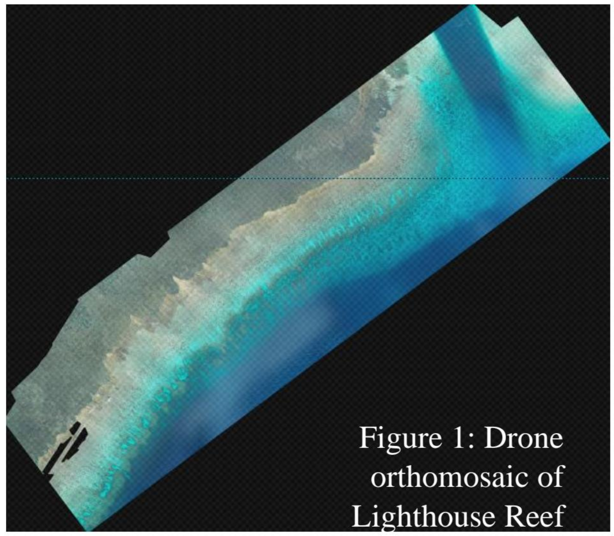

Figure 1: Drone orthomosaic of Lighthouse Reef

One of the challenges of this project was determining how to label our dataset. Ultimately, we chose to import the photomosaic into the illustrator program GIMP, and added a new layer where we manually colored in the regions we deemed to be coral (Figure 1 below). This new layer could then be used as our "ground truth" data for the model to train on. Determining precisely what was coral or not coral was challenging at times. For example, rocky substrate can look very similar to corals at the distance and resolution of the drone's imaging. It is worth acknowledging that human error may be adding bias to the model training. In the future, it would be preferable to use co-located underwater images similarly to Li et al. (2020) to better constrain our labels.

Once segmented, the images were saved in four sections (see example Figure 2). These were then split into squares of either 64^2 or 128^2 pixels for input to the model. 64^2 tiling resulted in 3,402 total images, while 1282 resulted in 3,402 labeled tiles with no overlap. These were then shuffled randomly and 90% taken as training data and 10% as testing data.

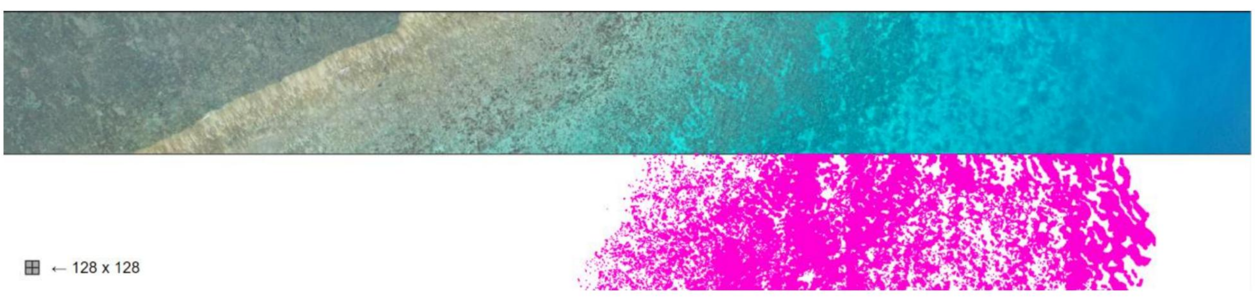

Figure 2: Labeled drone imagery for model training/testing. The original drone imagery (top) was used as model input and a binary segmented image of the same reef section (bottom) for model output. The square in the lower left corner shows the size of a 128x128 square, our training image size.

4 Methods

We used a modified U-Net model for our image segmentation algorithm (Weng & Zhu, 2021), following the segmentation tutorial on the Tensorflow website.[1] This model was selected as it is the same framework used by Li et al. (2020) and a common approach for rapid image segmentation. It consists of an encoder and decoder, which act (respectively) to down-sample the input image using convolutional filters and then up-sample the result with convolutional transpose operations until it is returned to its original shape. For layer in the down-sampling process, results from the layer's activation are also fed directly into the correspondingly sized up-scaling layer such that spatial information is preserved in the final model output. The loss function is a binary cross entropy loss. Figure (3) shows the model architecture. Our code can be found here: https://github.com/david-boles/cs1430-final-coral.

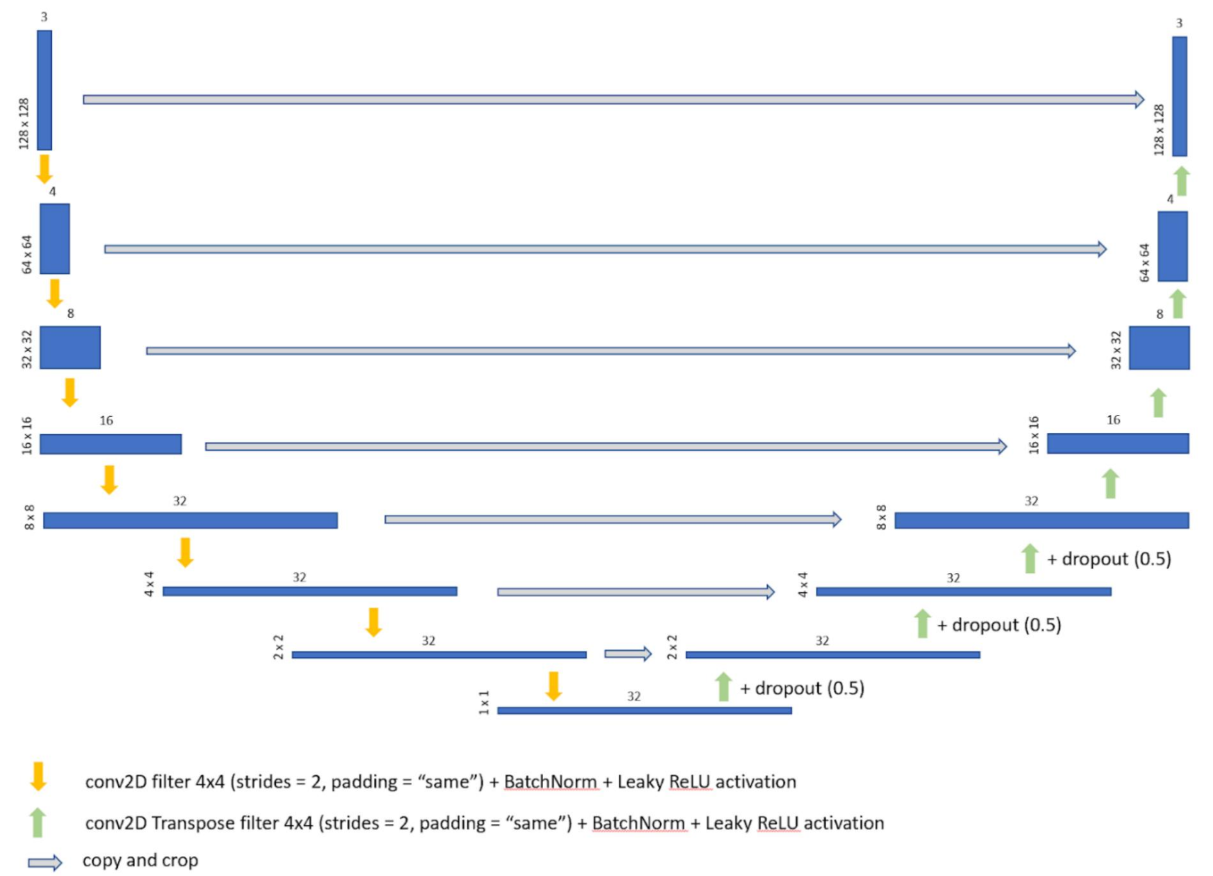

Figure 3: Schematic of the modified U-Net model architecture, for an input size of 128^2 and fewer filters.

5 Experiments/Results/Discussion

The network trained quite quickly (less than one minute per epoch on a CPU) and reached a peak validation accuracy of 85% with relatively minimal gains after 10 epochs.

Four different experiments were run to examine how changes to model complexity and input image size affected our results. To look at the effects of increasing model complexity, we

[1] https://www.tensorflow.org/tutorials/images/segmentation

doubled the number of filters in each layer compared to what is depicted in Figure (X). We examined how the model performed with input images sized 64x64 pixels compared to 128 x 128 pixels.

We found that neither changes to model complexity or input image size had much effect on the metrics of the final model output. All of the models reached an accuracy threshold around 94% after about 10-15 epochs. Precision and recall, perhaps more useful metrics for this case given that many of the training images had no coral present in them, reached about 85%. The models with the smaller input images produced precision and recall values 1% and 2% higher than corresponding large image models after 15 epochs. I expect that this small improvement in performance is due to having a larger volume of input images. Increasing model complexity improved training accuracy slightly (a couple percentage points) but did not affect validation accuracy, so this seems to be the optimum accuracy for this network structure.

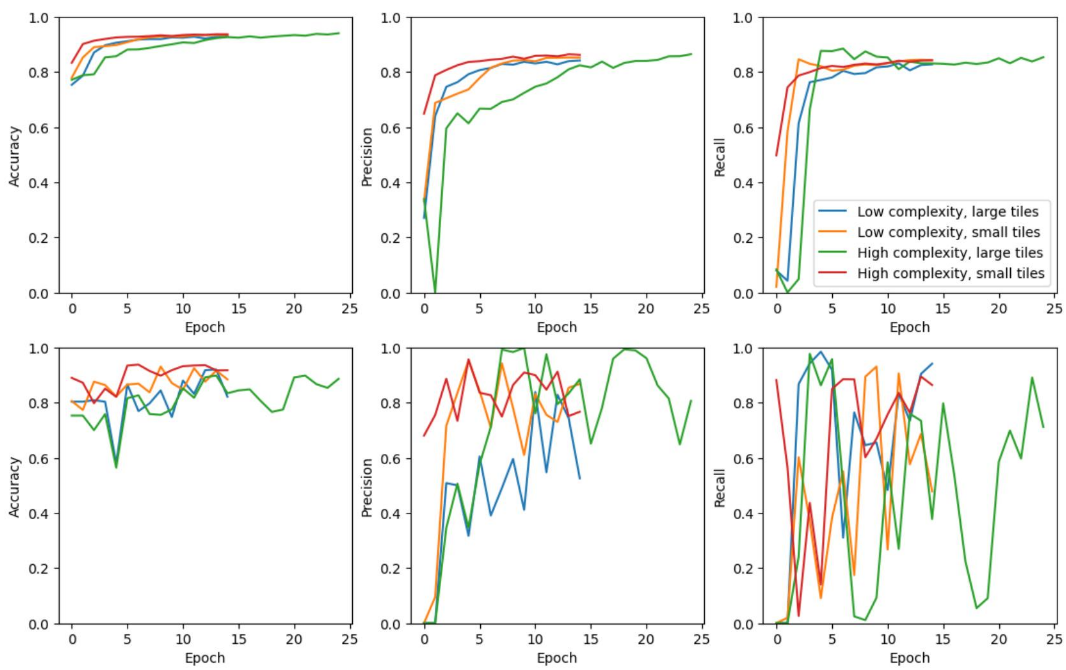

Figure 4: Model metrics (accuracy, precision and recall) after each epoch of testing. Top row is for training data, bottom row is testing data. The test data are all over the place; I realized putting this report together that I think I calculated them for each batch and not for the entire test set, which is a small set of images and likely results in such scattered results. Unfortunately I ran out of time to correct this error.

Table 1: Training metrics for each model after 15 epochs. Validation metrics were not included (see explanation in caption of Figure 4).

Model	**Loss**	**Accuracy**	**Precision**	**Recall**
Low complexity, large image	0.93	0.16	0.84	0.83
Low complexity, small image	0.93	0.15	0.85	0.84
High complexity, large image	0.93	0.16	0.85	0.82
High complexity, small image	0.94	0.14	0.86	0.84

6 Conclusion/Future Work

Given the uncertainty in our human labeling, the results of the model are quite remarkable. Looking at the segmented tile outputs, it appears that there are cases where the model identifies coral more accurately than our input labeled set. It is noteworthy that the model performed well at all depths on the coral reef, identifying individual coral bommies in the shallower regions and picking out the darker shadows of coral masses that fade into the background in deeper regions (Figure 5). We assembled a full reef mosaic of the model outputs (using the less complex, large images model) and the results are quite stunning, and show clearly that the model is able to capture the main sections of the image that contain coral and those that do not. The model may be a little over predicting coral cover, but the dramatic reef "spur and groove" topography is particularly clear in the outputs.

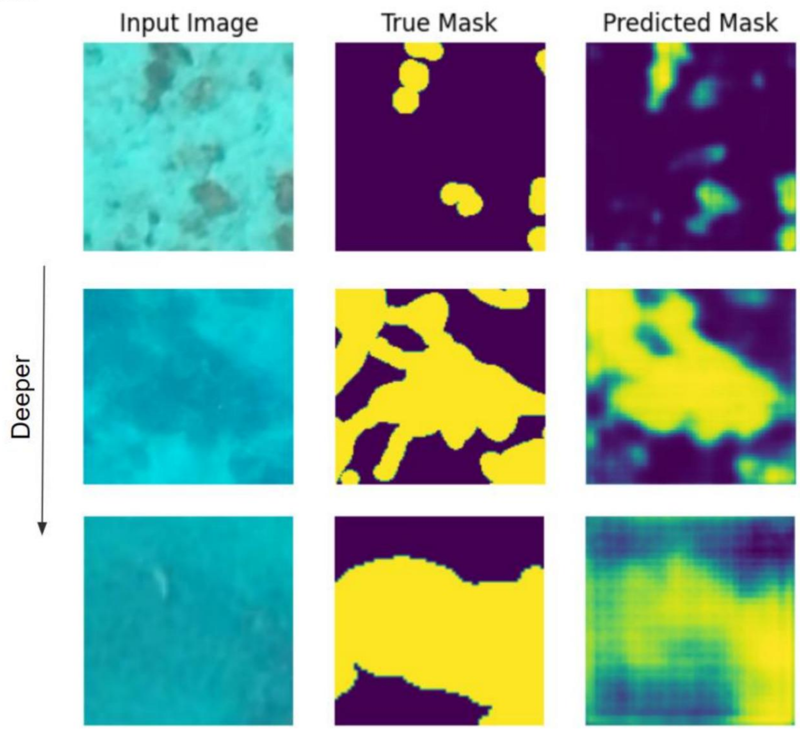

Figure 5: Model predictions on individual tiles (right column) compared to input images (left column) and our labeled images (middle). The top row shows a shallow region of the reef flat, the bottom row one of the deeper regions.

There are several directions I would like to take this coral identification model going from here. More work can likely be done to improve the performance of the model. I suspect that it would benefit from having more training data, but we ran out of time this quarter to do more image labeling. Underwater imagery taken on the reef could also be co-located with the drone photogrammetry and substantially improve human labeling accuracy. I suspect having a better "ground truth" dataset would be the most effective way of improving the model results.

I would also like to develop the model further to incorporate other classes of benthic cover, such as macroalgae, sand, rock and seagrass. Additionally, it would be interesting to evaluate the model on drone imagery from other reefs to assess whether weights learned in one region can be easily applied elsewhere.

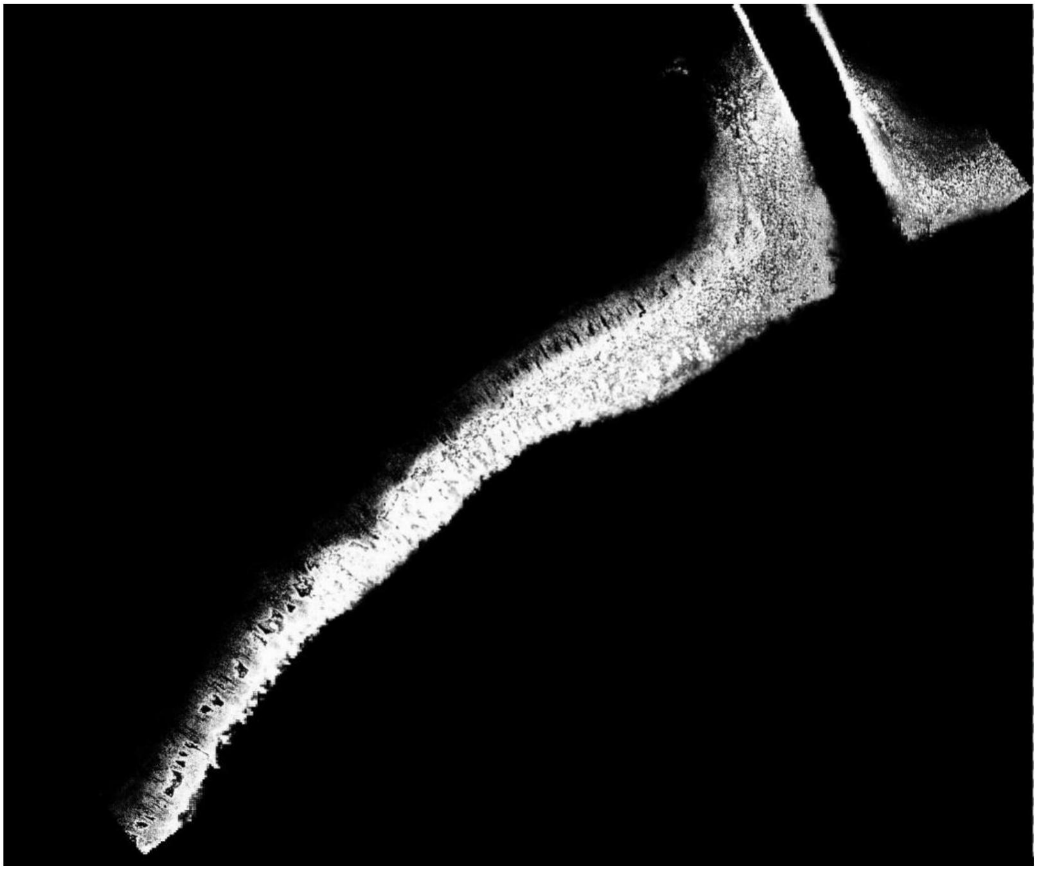

Figure 6: Full reconstruction of drone mosaic from model outputs. White represents regions the model has identified as coral, while black is non-coral.

7 Contributions

Both David Boles and Elisa Boles contributed to manually labeling the dataset. David wrote the base model script, and Elisa performed the tests changing various parameters, evaluated the metrics, and wrote up the final report.

8 References

Gonzalez-Rivero, M., Beijbom, O., Rodriguez-ramirez, A., Bryant, D. E. P., Ganase, A., Gonzalez-marrero, Y., et al. (2020). Monitoring of Coral Reefs Using Artificial Intelligence : A Feasible and Cost-E ff ective Approach, 1–22.

Li, A., Chirayath, V., Segal-rozenhaimer, M., Torres-perez, J. L., & Bergh, J. Van Den. (2020). NASA NeMO-Net ' s Convolutional Neural Network : Mapping Marine Habitats with Spectrally Heterogeneous Remote Sensing Imagery. https://doi.org/10.1109/JSTARS.2020.3018719

Weng, W., & Zhu, X. (2021). UNet: Convolutional Networks for Biomedical Image Segmentation. *IEEE Access*, *9*, 16591–16603. https://doi.org/10.1109/ACCESS.2021.3053408

Williams, I. D., Couch, C., Beijbom, O., Oliver, T., Vargas-Angel, B., Schumacher, B., & Brainard, R. (2019). Leveraging automated image analysis tools to transform our capacity to assess status and trends on coral reefs. *Frontiers in Marine Science*, *6*(APR), 1–14. https://doi.org/10.3389/fmars.2019.00222