Spring 5-22-2019

Using Computer Vision to Quantify Coral Reef Biodiversity

Niket Bhodia
San Jose State University

Using Computer Vision to Quantify Coral Reef Biodiversity

A Thesis Project

presented to

the Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Niket Bhodia

May 2019

The Designated Project Committee Approves the Project Titled

Using Computer Vision to Quantify Coral Reef Biodiversity

by

Niket Bhodia

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

San José State University

May 2019

Dr. Philip Heller	Department of Computer Science
Dr. Thomas Austin	Department of Computer Science
James Casaletto	University of California, Santa Cruz

ABSTRACT

Using Computer Vision to Quantify Coral Reef Biodiversity

By

Niket Bhodia

The preservation of the world's oceans is crucial to human survival on this planet, yet we know too little to begin to understand anthropogenic impacts on marine life. This is especially true for coral reefs, which are the most diverse marine habitat per unit area (if not overall) as well as the most sensitive. To address this gap in knowledge, simple field devices called autonomous reef monitoring structures (ARMS) have been developed, which provide standardized samples of life from these complex ecosystems. ARMS have now become successful to the point that the amount of data collected through them has outstripped the capacity of research organizations to analyze through molecular methods. To facilitate these efforts, the present study explores the use of computer vision techniques to analyze the complex image data of these samples in order to extract useful information based on morphological (visual) characteristics of the collected organisms. Various techniques at varying levels of sophistry are surveyed for their suitability to the present problem. In the end, the more complex techniques are ruled out in the favor of basic image processing ones, of which three are tested: canny edge detection, color space transformations, and histogram equalization. While the first one does not directly yield useful results, the latter two turn out to be surprisingly effective, showing great promise as means to prepare data that more sophisticated techniques can be subsequently trained on. Future directions of investigation are recorded in detail, along with suggestions and relevant references, towards ultimately realizing an online analysis tool and repository for marine life that would accelerate related research and conservation efforts.

Index Terms—**computer vision, OpenCV, deep learning, machine learning, unsupervised learning, artificial neural networks, image processing, autonomous reef monitoring structures, coral reef, biodiversity, Census of Marine Life**

Acknowledgements

To Dr. Philip Heller, for being Yoda to this young padawan (enough said);

To Dr. Thomas Austin, for being unfailingly patient, generous, and cheerful, as well as for peppering his lecture slides with geek humor;

To James Casaletto, for bringing so much energy to 7.30 am lectures and doing it with such style;

To Dr. Mark Stamp, Dr. David Taylor, Dr. Chris Pollett, Dr. Guha Jayachandran, Dr. Volkmar Frinken, Shriphani Palakodety, Kevin Smith, and Ramin Moazzeni, for enabling the best educational experience of my life during this Master's course;

And to Dr. Sami Khuri, for the open CRISPR course last summer, which deepened my appreciation for bioinformatics and led to me being part of this project.

Table of Contents

List of Figures

1. INTRODUCTION

Common knowledge has it that Earth's forests produce the oxygen that sustains so much of the life on the planet. However, the fact is that ocean life, specifically microscopic organisms called *phytoplankton*, produce as much oxygen as their terrestrial counterparts [1]. In addition, the combined actions of certain ocean organisms constitute the biological pump that sequesters ~40% of anthropogenic carbon emissions [2], mitigating climate change. Ocean life is also an important source of food, economic activity, and medicinal drugs [3], [4]. Hence, understanding ocean life and how it is impacted by human activity is a question that pertains to our very survival.

Arguably, the most interesting of ocean habitats are coral reefs. Often referred to as the *rainforests of the sea*, they occupy only 0.2% of the ocean surface, yet are estimated to host around 35% of all marine species [5]. They are also exceptionally sensitive to human impact, the proverbial canary in a coal mine warning us about the irreversible damage we are doing to the planet. In 2008, it was estimated that 60% of all coral reefs were degraded or lost [6] and that one-third of all corals were at the risk of extinction, making them the most endangered group of animals on the planet [7]. Around the same time, another study predicted that given current trends in greenhouse gas emissions, coral reefs will cease to exist meaningfully by 2050 [8].

These factors make coral reef biodiversity an urgent topic of study. Biodiversity is an important metric because it increases ecological resilience by introducing functional redundancies within ecosystems [9]. However, before we measure biodiversity loss in coral reefs, extant biodiversity levels first need to be quantified. Current estimates [5], [10], [11], while valuable efforts, are only extrapolations. They put the number of species hosted by coral reefs in the range of 500,000 to 10 million, with even the lowest bound in that range far exceeding the number of species described so far. It is estimated that 85% to 99% of coral reef species still remain undescribed [9].

Fig. 1. Image of a coral reef, attesting to the abundance and variety of life these habitats host.[1]

[1] Reproduced from the BBC America website under fair-use guidelines. Link: http://www.bbcamerica.com/shows/planet-earth-blue-planet-ii/season-1/episode-03-coral-reefs

Bridging this gap in knowledge poses several challenges, apart from the fact that data collection in ocean environments is inherently resource intensive. The most effective means of data collection for visible species is hand sampling, but this is an unscalable approach given the magnitude of diversity and the high degree of species endemism in coral reefs, whereby many species are unique to a certain limited geographic region and are found nowhere else in the world. Further, divers vary dramatically in their ability to collect samples, which makes standardization difficult. Finally, removal of samples from coral reefs constitutes destructive sampling of already impacted habitats, an anathema to most marine researchers, and requires securing permits from authorities [9].

Given this background, the use of devices called autonomous reef monitoring structures (ARMS) has emerged as a key methodology for standardized sampling of coral reef ecosystems. ARMS were developed under the Census of Coral Reef Ecosystems project, one of the 14 field projects comprising the Census of Marine Life, a 10-year international effort that involved 2,700 scientists from 80+ nations [12]. Each ARMS consists of eight gray 23 cm × 23 cm PVC plates stacked in alternating open and partitioned layers, the latter created using X-shaped inserts to divide the space between the plates into four sections. The stack is topped by a plastic pond filter mesh and a final plate, and is mounted on a 35 cm × 45 cm base plate. The structure is deployed at reef sites for extended periods, typically 1 to 3 years, where they somewhat mimic the structural complexity of reef structures and end up becoming habitats for sessile (stationary) and motile (mobile) organisms. They are not designed to capture all or even the maximum

number of species, but rather to collect important indicators of diversity [9], [13], particularly sessile colonizing invertebrates.

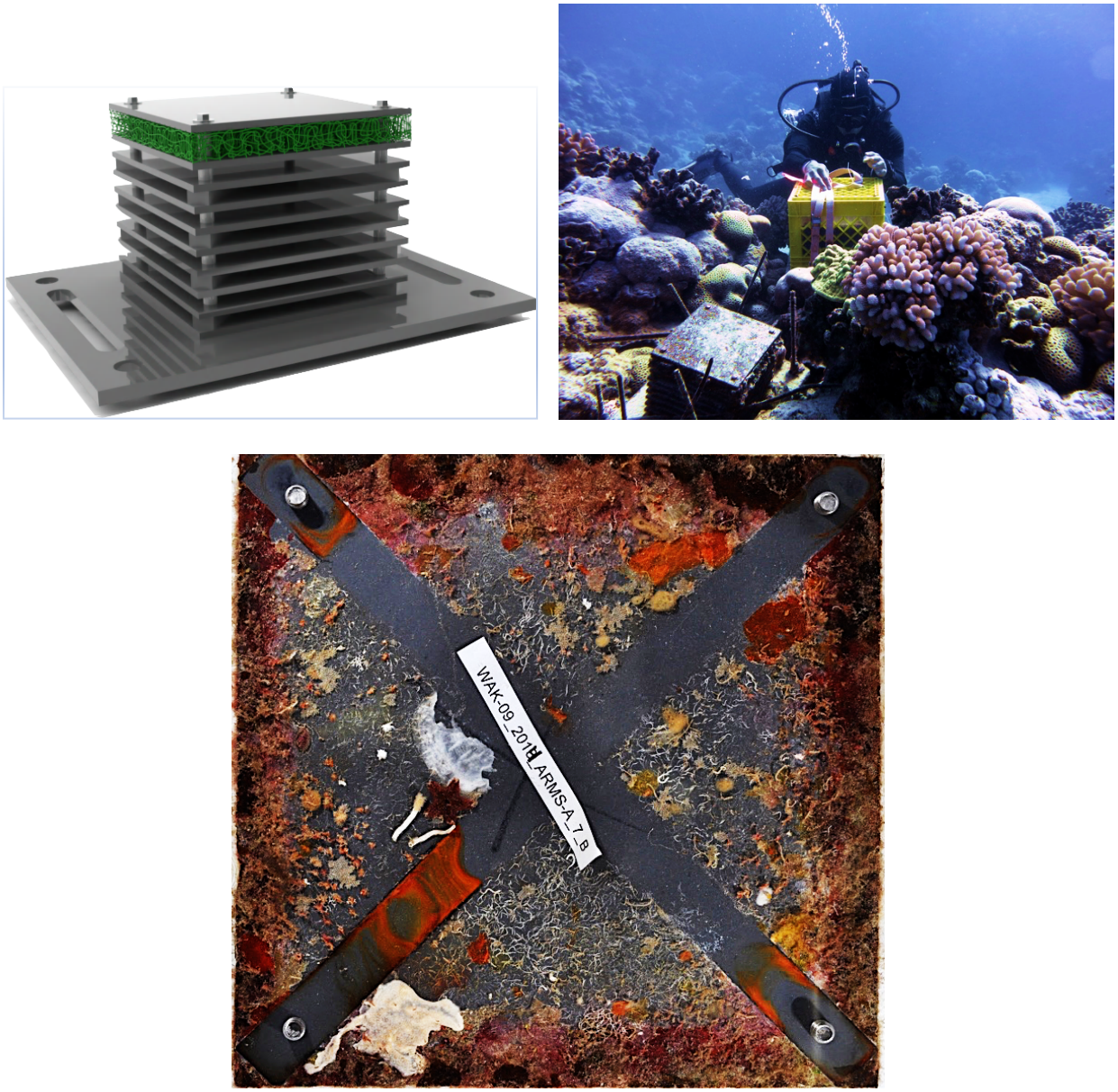

Fig. 2. (Top left) Autonomous reef monitoring structure (ARMS).[2] (Top right) Diver retrieving an ARMS off Wake Island, Micronesia, Pacific Ocean. (Bottom) Photograph of one of the plates obtained from the same diving expedition, taken after the retrieved ARMS was disassembled.

[2] Obtained from informational material provided by the *Census of Coral Reef Ecosystems* (CReefs).

Simple, rugged, and cost-effective, ARMS have been adopted as a standard sampling approach by research institutions worldwide. They have been an important component of National Oceanic and Atmospheric Administration (NOAA)'s Pacific Reef Assessment Monitoring Program, NOAA's Biodiversity Alternative, and the Smithsonian's Marine Initiative [9]. As of 2015, there were over 1600 ARMS deployed worldwide with huge investments by several organizations and ongoing efforts to standardize processing protocols to make the results globally comparable [13]. With ARMS now being deployed to marine habitats other than coral reefs, the number is only set to grow [14].

ARMS efforts have now reached a point where biodiversity assessment attempts are confronted with another typical big data bottleneck: analysis of collected data [15]. Genomic analysis is the ideal approach for analyzing the collected ARMS samples, but both techniques that fall under it have limitations. The first, genetic barcoding, relies on species identification based on a specific genetic subsequence, rather than the entire genome. However, this requires sampling individual organisms from collected material, which is very resource intensive. The second, metagenomics, is based on sequencing all genetic material from a sample of collected material and then identifying and quantifying the organisms within the mixed sample. However, currently, this technique presents a few unresolved challenges, especially for detecting multicellular life [9], [16].

This leaves visual analysis as perhaps the only remaining alternative. Visual identification of collected organisms by experts is unfeasible for several reasons: lack of

scalability, lack of taxonomic expertise, the existence of cryptic species (different species that look highly similar), and the inherent unreliability of identifying and classifying new species based on morphological (visual) characteristics. However, if the process of analyzing photographs of the retrieved ARMS plates can be automated reliably, then these drawbacks could be fruitfully addressed to various degrees.

This is where computer vision (CV) can help. CV is the branch of computer science that deals with algorithms, techniques, and approaches that enable computers to extract useful information from images. The field has gained prominence in recent years due to the extraordinary success of supervised deep learning (DL) in CV applications. However, DL is just one of several machine learning (ML) techniques that can be applied to CV, and CV, on the other hand, employs many techniques outside of ML. All these techniques have utility depending on the use case, which we discuss next.

2. BACKGROUND

In this section, we analyze the problem and assess the applicability of various computer vision (CV) techniques to address it. The flow of this section mirrors the thought process Dr. Heller and I went through as the project progressed.

Let us first consider the nature of the problem. Fig. 3 shows another example of a retrieved plate. This is a good example because you can see the variety of organisms a single plate can hold. You can have red- or orange-colored patches of microorganisms, pulpy yellow or orange blobs, white noodle-shaped creatures, plant-like growths, and shelled specimens. The Heller Research Group has a dataset of nearly 15,000 such images courtesy of Dr. Russell Brainard, NOAA's Supervisory Oceanographer for coral reef science. The images are from 17 sites in the Pacific Ocean for ARMS recovered from 2013 through 2015.

It is important to note that not all these images are those of retrieved plates—some only record the location where they were recovered, and the diving effort undertaken to do so. Even among the plate images, not all are unique. Each plate has been photographed multiple times. With images from 17 sites, an average 3 locations per site, an average 3 installations per location, an average 2 ARMS per installations, and 8 plates per ARMS, the dataset is estimated to contain around 2,500 unique images. This estimate is

corroborated by the observation that there are, on average, 5-6 images for every unique plate. And these are just a fraction of the ARMS images that have been collected worldwide. Oceanographers, ecologists, and marine biologists would ideally like a system that could extract the different organisms from these images, classify them correctly, and store them in a globally accessible database.

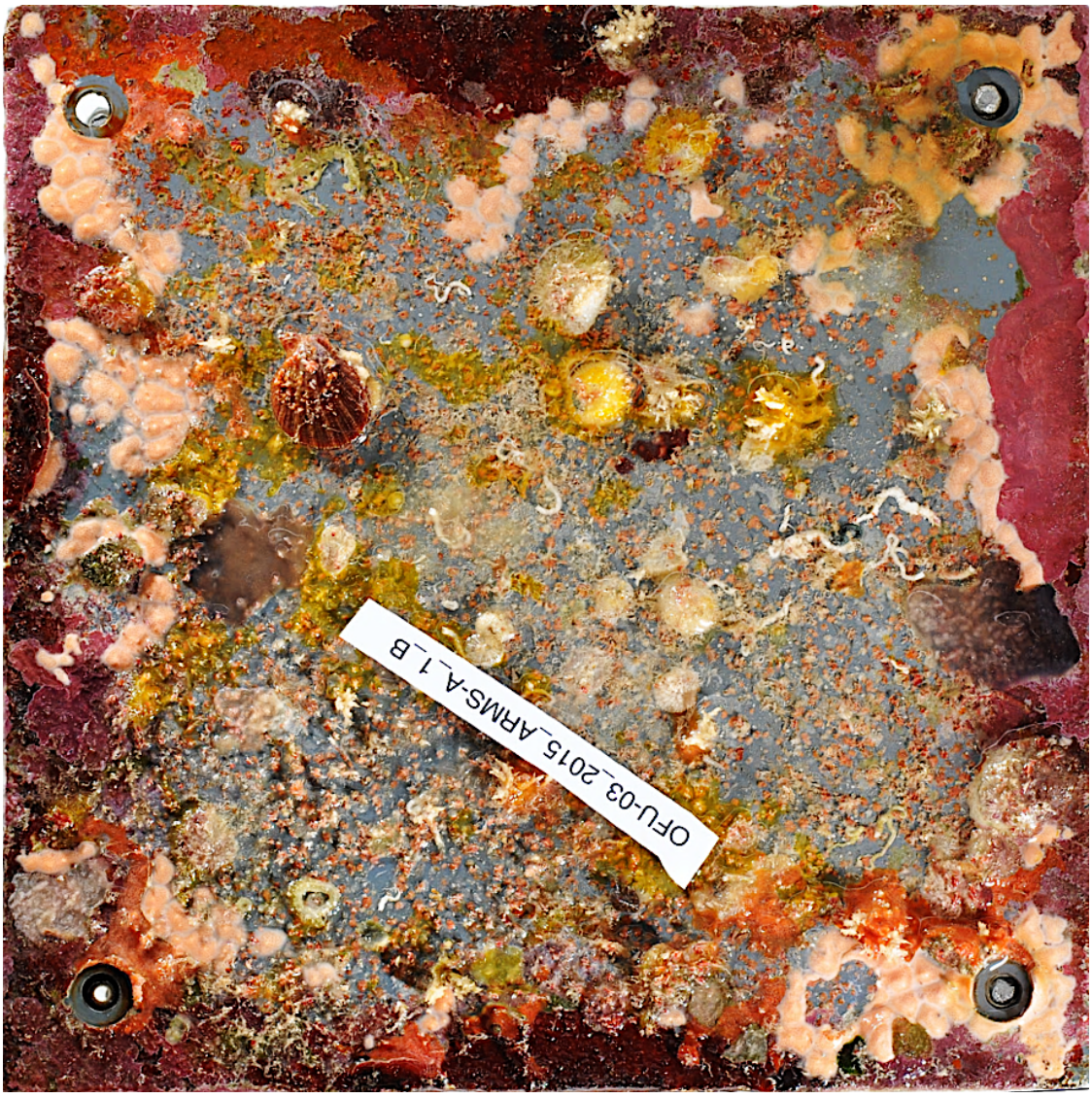

Fig. 3. Example ARMS plate retrieved off the coast of Ofu in the Samoan Islands, Pacific Ocean.

For a task of this magnitude, our natural instinct would be to first go for the best technique available in CV—deep learning (DL)—and that is what Dr. Heller and I initially conceived the project to be about. DL is a machine learning technique that involves training artificial neural networks (ANNs)—computational models that mimic the functioning of the human brain—over several iterations or *epochs*. The "deep" in "deep learning" comes from the fact that the ANNs trained with DL algorithms have multiple "hidden" layers, anywhere from 2 to 1000, which allows the ANNs to automatically extract hierarchical representations based on which they can approximate complex relationships. Interested readers can refer to [17][18] for more details.

DL sparked intense interest in 2012, when it made breakthrough progress in the ImageNet Challenge, beating the competition by a considerable margin. Since then, it has surpassed humans on the ImageNet Challenge [19], and delivered breakthroughs in a wide variety of tasks, including problems in wildlife and marine biology. For instance, Norouzzadeh et al. [20] used DL to automate the identification, counting, and description of wild animals in camera-trap images, achieving a 96.6% accuracy as compared to human volunteers while saving 99.3% of the effort. Sun et al. [21] were able to employ pretrained DL models to achieve a precision of 99.68% and a recall of 99.45% when classifying between 26 species in low-quality underwater videos. Mahmood et al. [22] have employed DL for the automatic annotation of coral species in reefs with up to a 97% accuracy from imagery collected by autonomous underwater vehicles.

However, these DL successes have been achieved in the context of supervised learning, wherein the ANNs are trained using manually annotated datasets, such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset [23] (approx. 1.2 million images spanning 1,000 classes) or the COCO Dataset [24] (approx. 200,000 images textual captions and object segmentation information). These are images of objects and living beings we encounter in everyday life that bear no correspondence to the reef organisms collected with ARMS. Furthermore, as evidenced by the sizes of the ILSVRC and COCO datasets, supervised learning requires massive amounts of such annotated data. Building such a massive dataset with, say, 1,000 examples per each class all annotated by human experts when so much of the sea life is unknown is a task bigger than the problem itself. Even if we take into account transfer learning, whereby networks pretrained on massive datasets like ImageNet or COCO are then honed on smaller, specialized datasets to yield good results for niche applications, the undertaking would be beyond our reach.

Alternatively, we could resort to unsupervised applications of DL. Assuming the different organisms have been extracted from each plate, they could be clustered together based on morphological similarities, rather than classifying the data into the correct category based on a labeled supervised learning dataset. Each cluster could then be studied, quantified, and/or annotated by the appropriate experts, accelerating research in the respective areas. Such DL methods have been reported, with impressive results. For example, in [25], the authors consider the chicken-and-egg problem of how good representations are required to achieve good clustering results, and conversely, how

good clustering results allow us to refine the representations. They propose an iterative scheme, wherein image clustering is conducted in the forward pass and representation learning is performed in the backward pass, resulting in a robust model that generalizes well across various tasks and datasets. The authors have made the implementation publicly available, which makes for a promising avenue in terms of future work.

However, that is a later consideration as the different organisms still need to be extracted from the plate images. This becomes a segmentation task, specifically, boundary segmentation, which once again needs to be unsupervised because of lack of labeled data. The organisms do not have fixed shapes, and human vision need to rely on a combination of color, texture, depth perception, and learned experience to tell these organisms apart. Unsupervised CV would need to rely on just color and texture, of which unsupervised texture-based segmentation is the real technical challenge. Some impressive work has been reported in this area last year [26], wherein the authors use a two-step process. The first involves using a randomly sampled subset of patches from an image to train convolutional features using an objective function and certain constraints. The trained convolutional features have the property that they show low variation across the same texture but a high variation at the edges of texture boundaries. This stage is computationally intensive and is made tractable by introducing certain relaxations to the equations. The second step is to then use these filters to segment the image based on a metric called the Mahalanobis distance. This is followed by a post-processing stage that

merges small spurious regions into larger ones. The authors successfully applied this method to prepared sample images and actual histology images.

However, the math underlying this technique is fairly sophisticated, and its implementation requires several optimizations to make the computations feasible. The source code is not publicly available, so I contacted the corresponding author to inquire whether the authors would be willing to share it for us to test on ARMS data. I have received no response yet, but perhaps another attempt can be made in the future given that ARMS is an ideal case study for the technique and the results should be interesting for the authors too.

Of course, we could use other, simpler ML techniques to carry out the segmentation. This was attempted on the ARMS data last year wherein the author used the *k*-means and the fuzzy *c*-means (FCM) clustering algorithms to segment the images by clustering pixels in the color space [27]. The *k*-means algorithm is one of the simplest ML algorithms. We start by assuming that k clusters exist and randomly initialize k corresponding cluster centroids in a Euclidean space with d dimensions in which our n data points are plotted. Then, every point is assigned to its nearest centroid. Once all points are assigned, the centroids of the resulting clusters are calculated, and the original centroids are updated with these newly calculated centroids. This continues for i iterations until convergence is reached. FCM is based on the same concept, but with one key difference: while in *k*-means, every point can belong to only one cluster (hard assignment), FCM tracks the

probability of each point belonging to each of the clusters (soft assignment). The probability is assigned based on the distance of the point from the cluster centroid.

The limitation of the implementation in [27] was that it only considered the color space and did not take the textures or spatial relationships into account. Consequently, a single, more complex organism showing different colors would be segmented across multiple clusters, as shown in Fig. 4. The same would also occur for a single group of organisms (the red and orange patches in Fig. 4) due to minor changes in color, which complicates matter further downstream as these clusters would need to be recombined subsequently at some point. Conversely, different organisms would be grouped into the same cluster simply because they share similar colors, as Fig. 4 illustrates.

One could perhaps improve on this by extracting texture descriptors and clustering in spatial-color-texture space [28], but this will increase the number of dimensions and incur a performance penalty given that the time complexities of k-means and FCM are $O(nkdi)$ and $O(nc^2di)$, respectively, where n is the number of points; k and c, the number of clusters; d, the number of dimensions; and i, the number of iterations required for convergence. Moreover, say, we started with 2,000 images, each yielding 5 clusters on average. Once they all are segmented, n = 10,000; all of these will need to be reclustered. And, practically speaking, even the c and i values will increase when so many organisms are put together in a single bucket, and as more organisms are added subsequently.

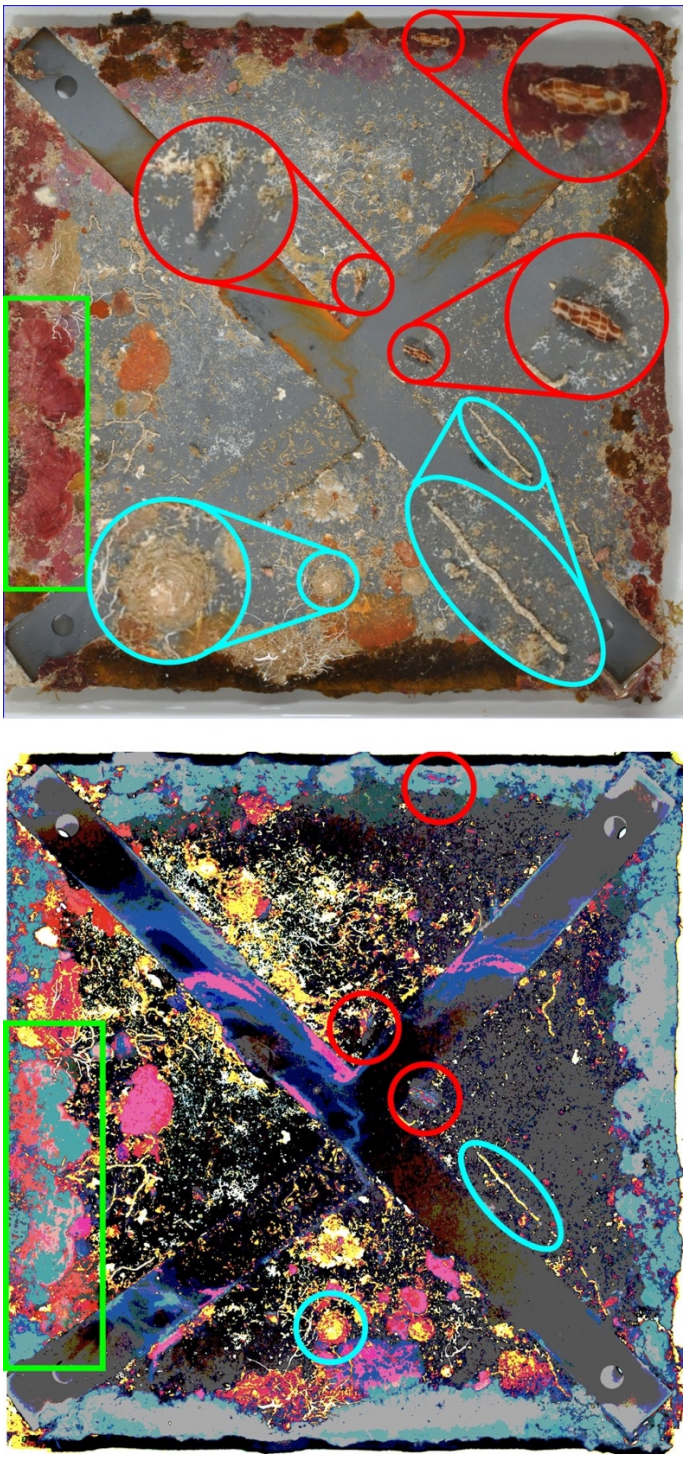

Fig. 4. (Top) Original image. Red circles, blue circle, and blue ellipse indicate complex specimens, zoomed in for clarity. Green rectangle indicates single group of red organisms discussed later. (Bottom) Image segmented using fuzzy c-means (FCM) clustering with the implementation in [27]; the overlaid colors indicate the clusters the pixels have been grouped in. Note how the complex specimens marked by red circles are divided across different clusters. Same occurs for the colony of red organism marked by the green rectangle, due to minor color variations. Blue circle and ellipse indicate two different organisms grouped into same cluster because of similar colors. FCM showed better results as compared to k-means, which is why the comparison is being carried out with an example from this method.

Further, while both algorithms guarantee convergence, they do so only to local minima. Hence, the recommended approach is to perform a number of random restarts and retain the best result. The methods also need to be supplied with a k or c value, which there is no way of knowing beforehand for each plate with the ARMS data. The common approach to dealing with this problem is to perform repeated runs assuming k or c as 1, 2, 3 …, computing the clusters for each k or c value. The average distance to centroid a is tracked till the *elbow point* can be determined, that is, the point in the k (or c) vs. a graph until which a decreases sharply and subsequently slows down. This elbow point is considered the correct k value.

The time complexity presented earlier does not account for the restarts and the elbow method—incorporating these makes the time function grow as $O(nk^2dir)$ and $O(n(c^2)^2dir)$, where r denotes the number of restarts. Optimized implementations do exist for k-means and FCM, which work or reducing repeated work since the points do not move between clusters too often. Their span can also be reduced through parallelized implementations, as was done in 2018 by Bharath Baiju, an undergraduate in the Heller Research Group. But put all the considerations together—extracting textures, mapping pixels to a spatial-color-texture space, and the potential performance issues—we might be better served looking at alternative methods.

Hence, we started our discussion with DL, but then by a process of elimination, have systematically worked our way down to the simplest algorithms in CV, which are related

to basic image processing. This study will demonstrate how these techniques turn out to be surprisingly potent and show great promise in helping us build datasets that the more sophisticated techniques discussed earlier can be trained on in the future.

3. METHODS

Three key image processing techniques were tested: canny edge detection, color-space transformation to HSV space to separate hue and saturation ranges, and histogram equalization so that the same techniques and color ranges could be applied to a set of images. These operations were performed using OpenCV (v. 4.1.0), an open-source library with more than 2,500 classic and state-of-the-art CV and ML algorithms [29]. The library is written in optimized C++, which makes it computationally efficient and capable of meeting the needs of real-time CV applications. It has bindings in Python, Java, MATLAB, as well as wrappers in various other languages.

The code was implemented in Python (v. 3.6.8) using Jupyter Notebook (v. 5.7.8) on a Mid-2018 MacBook Pro running macOS Mojave 10.14.4 and equipped with a 2.6 GHz Intel Core i7 (8850H) processor, 16 GB of RAM, an AMD Radeon Pro 560X GPU with 4 GB of dedicated memory, and an Intel UHD Graphics 630 on-board GPU. Numpy, the popular array-processing package, was used as a prerequisite for OpenCV, and Matplotlib's Pyplot interface was used for plotting. Appropriate color ranges were identified with the help of the *Just Color Picker* desktop application [30] and *Alloy UI*'s online HSV Palette color picker [31].

4. RESULTS AND DISCUSSION

4.1 Canny Edge Detection

To address the limitations of *k*-means and fuzzy *c*-means, I first wanted to see whether organisms with coherent and regular shapes could be detected. If these could be removed from the images beforehand, then organisms that have no discernible shapes could subsequently be extracted using other techniques. To this end, the plan was to first detect the edges in the image, and then use these edges to find approximate shapes in the image. For edge detection, the images were converted to grayscale, Gaussian blurring was applied to remove high-frequency noise, and then OpenCV's built-in Canny edge detection function was used.

Canny edge detection is based on the use of the Sobel kernel, which is used to filter the image in the horizontal and vertical direction to obtain the edge gradient and direction for each pixel [32]. The OpenCV Canny function takes two arguments besides the image itself, the minVal and maxVal. Any edges with an intensity gradient below minVal are considered to not be edges and are discarded, while those with an intensity gradient above the maxVal are considered to be sure edges and are retained.

Experiments were run with various values of minVal and maxVal, with/without Gaussian blurring, and with/without histogram equalization. I also tried a function that

automatically sets the minVal and maxVal by first computing the median of the pixel intensities and then setting minVal = median * (1 - σ) and maxVal = median * (1 + σ), where $0 \leq \sigma \leq 1$ and is set by the user [33]. However, in all cases, the results were disappointing as the function returned very confused results, as shown in Fig. 5. I came to conclude that edge detection and contour finding (which depends on edge detection) might have a better chance if some other ways could be found to reduce the clutter on the plate first. Hence, I began looking at color, as discussed in the next section.

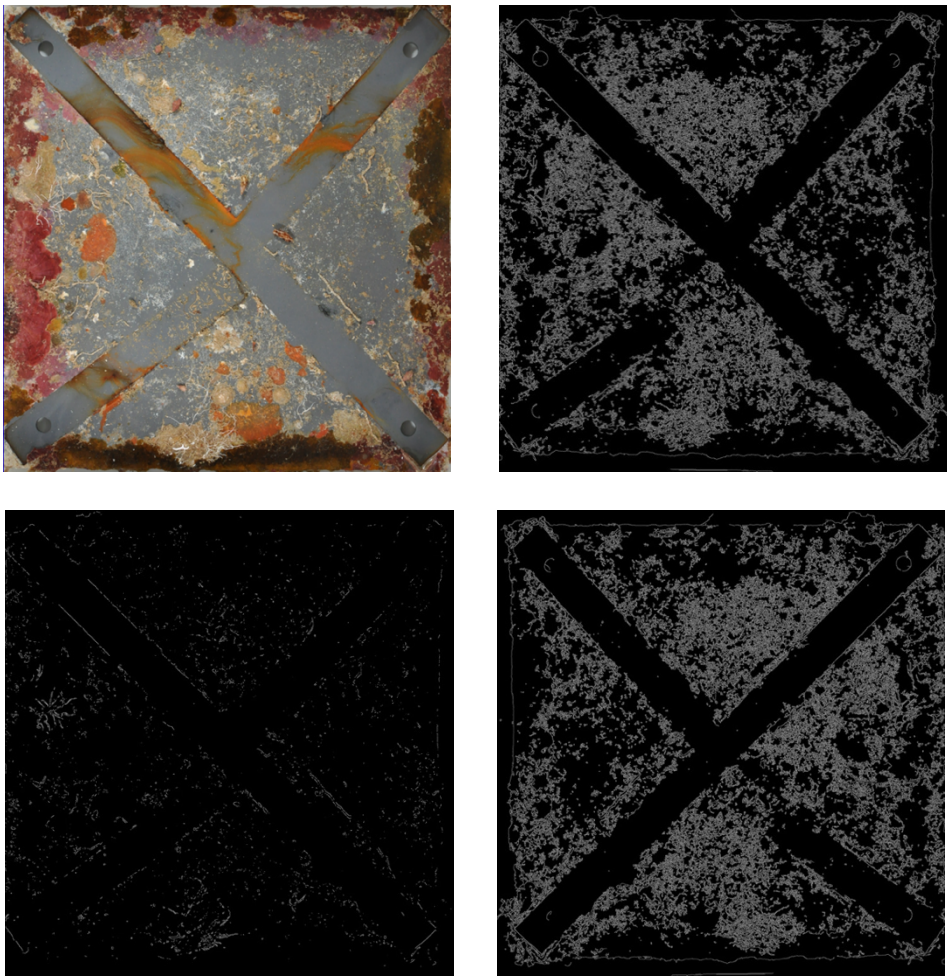

Fig. 5. (Top left) The original image. (Top right) Edges obtained by setting a wide range of minVal and maxVal. (Bottom left) Edges obtained by setting a narrow range of minVal and maxVal. (Bottom right) Edges obtained by automatically setting minVal and maxVal based on median pixel intensity.

4.2 Color Space Transformations

Consider Fig. 6, which shows a representative sample of the ARMS dataset.

Fig. 6. Sample images from the ARMS dataset. Notice the differences in the lighting conditions, the variation in plate population, the uniformity of the gray background, and the many plates having colonies of red-colored microorganism. The first image in this figure is the one we have been discussing in Fig. 4 and Fig. 5.

Notice the gray background of all the plates in this sample—setting aside the differences due to lighting conditions, the particular gray color is very uniform across plates. This is one of the consequences of ARMS standardization, and it could be used to our advantage—by subtracting this gray, some of the work might be made easier. In addition, one can see that several of the plates have colonies of red-colored microorganisms that look like rust. These are oxygen-producing organisms and are important to study and quantify; therefore, it would be useful to separate them from the rest of the information on the plate.

To carry out these tasks, we need a way to not just specify a single color, but a range of colors. The BGR (Blue-Green-Red) color space, which is default for OpenCV, and its more popular variant the RGB (Red-Green-Blue) color space, do not allow us to do that. But OpenCV allows us to convert from these color spaces to the HSV (Hue-Saturation-Brightness; also called HSB) color space with allows us to do exactly this. The HSV color space is frequently used in color-picker tools in graphic design software as it is intuitive to human users. The hue stands for the actual color, which is represented as a circle and takes values from 0° to 360° (although OpenCV scales the range down to 0° to 180°). The saturation represents how dark the color is, so assuming same lighting conditions a light blue might have the same hue but a lower saturation than a dark blue; it ranges from 0 to 255 in OpenCV. Finally, brightness, which too ranges from 0 to 255 in OpenCV, refers to how much light is shining on the object. Fig. 7 provides a visual representation of the RGB and HSV color spaces.

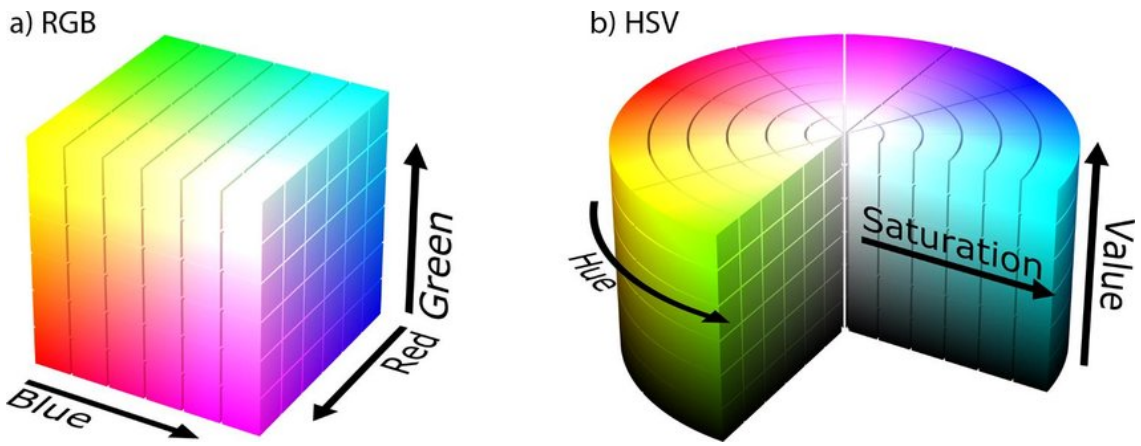

Fig. 7. An illustration of the RGB and HSV color spaces.[3]

HSV color space also makes sense intuitively in that all colors look like a certain shade of gray when not enough of that color is present on the surface and/or when not enough light is shining on it. This is perfect for our use case as it enables us to remove grays irrespective of which non-discernible hue they represent. In practice, I found that the grays of the ARMS plates had S <= 25%. The result of removing all pixels with S <= 25% is shown in Fig. 8 (top right). From this image, in turn, we can separate out the red patches. One can see in Fig. 8 (bottom right) how removing the grays and reds has done much to isolate the three shelled organisms in the image, making them more amenable to edge finding, contour finding, and texture detection. Note that generalizing these results will also require the brightness channel V to be taken into account, which is discussed in the Future Work section.

[3] Image by Michael Horvath, reproduced under the Creative Commons Attribution-Share Alike 3.0 Unported license.

On the other hand, the extracted red patches (Fig. 8 bottom right) also constitute important information and their data can be saved separately. Simply applying the procedure to a set of images and taking a pixel count of the reds gives us a datum that can be compared across the various ARMS sites. The red patches can also be sampled to create training data for future supervised learning efforts.

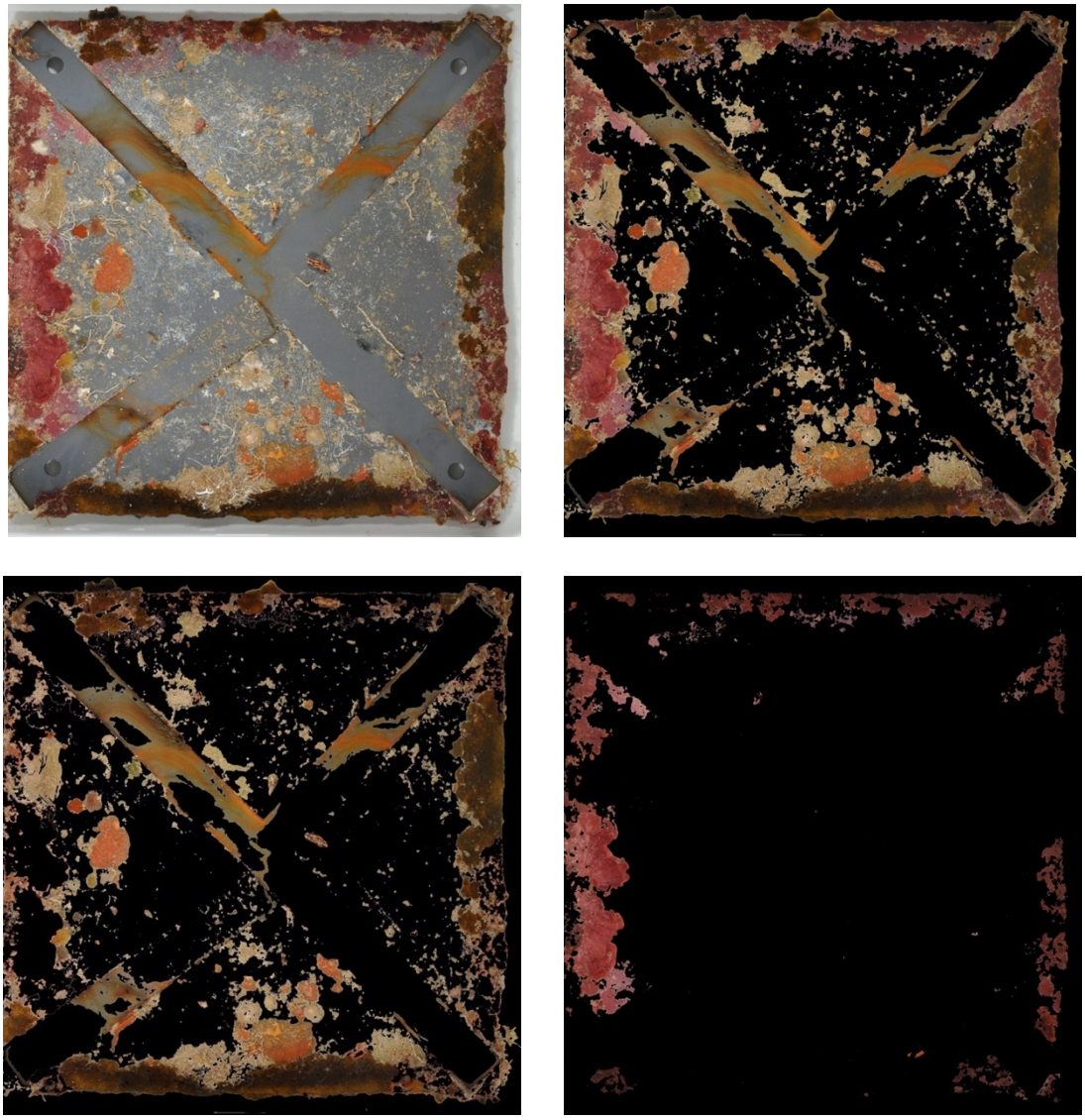

Fig. 8. (Top left) Original image, same as the one discussed in Fig. 4 and Fig. 5 (Top right) Image with the grays removed. (Bottom left) Image with the grays and reds removed. (Bottom right) The reds extracted from the top right image. These red patches can be analyzed further (e.g., by obtaining a simple pixel count to quantify them across images) or be used as training data.

4.3 Histogram Equalization

Examining Fig. 6 also shows us that the lighting conditions under which the ARMS images are captured vary substantially. We need some way to normalize the images so that the color space transformations and extractions can be applied uniformly to all images. In addition, improving the contrast in the images will certainly help researchers whenever they need to visually examine the images. Both concerns could potentially be tackled through histogram equalization.

In the context of images, histograms are graphical plots where the x-axis represents the discrete intensity values and the y-axis represents the number of pixels in the image with these intensity value. Often, the pixel intensities are confined to a particular range, making for poor contrast, and stretching out this range normally improves contrast. It can also potentially help us set uniform color and brightness ranges.

I tested two methods for histogram equalization in OpenCV. The default method depends on the global contrast of the image [34], the results for which are shown in Fig. 9. It is apparent from the results that this technique does not produce a visually desirable result. Notice too, how the original histogram was well spread out, and equalization nearly flattened it. Typically, this equalization technique only works when intensity values are confined to a certain narrow range; this is not typically the case for the ARMS images.

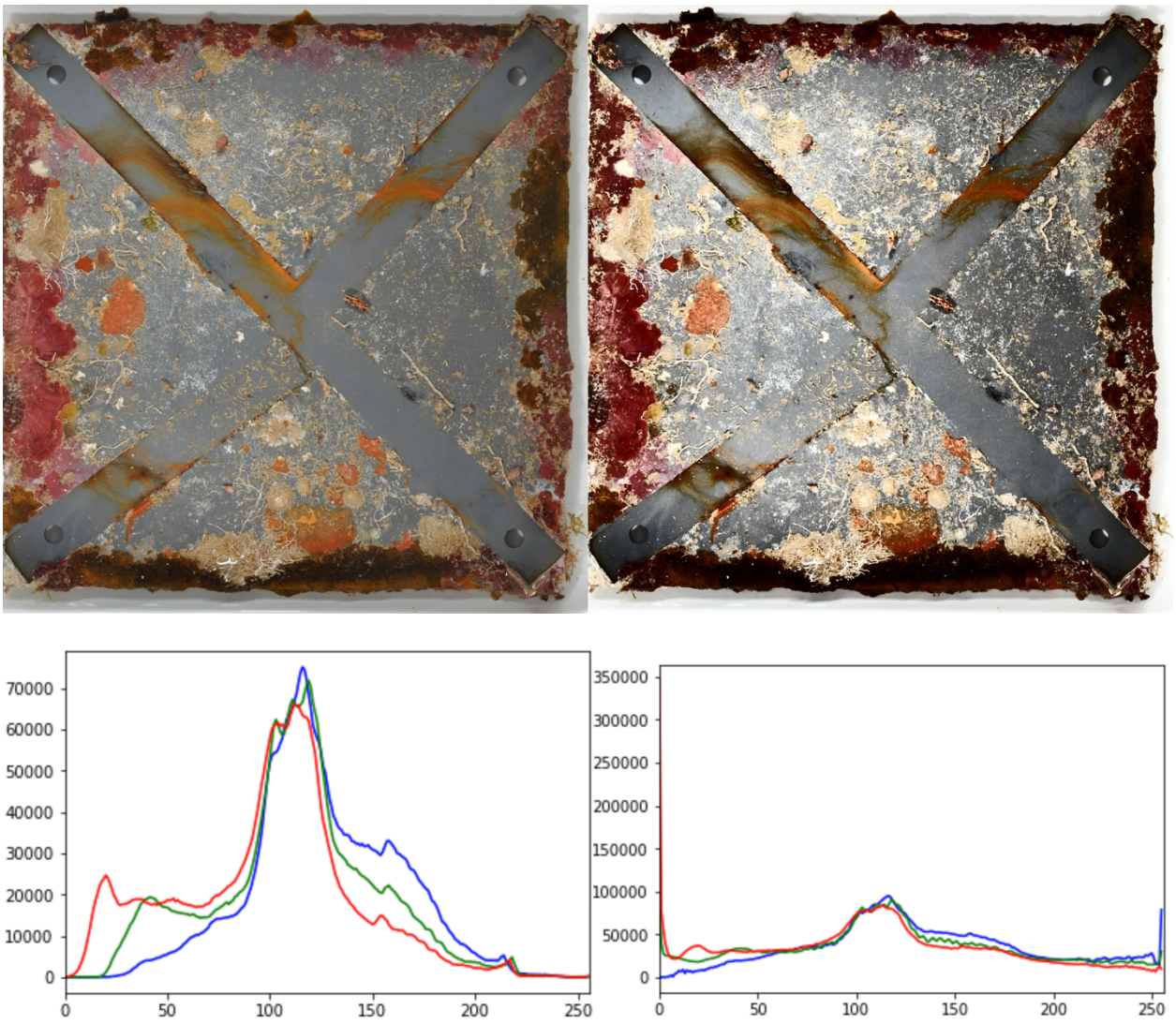

Fig. 9. (Left) Original image and its corresponding histogram. (Right) Image equalized based on global contrast and the resulting histogram. The red, green, and blue lines correspond to the intensity distribution in the red, green, and blue image channels, respectively.

I therefore experimented with contrast-limited locally adaptive histogram equalization (CLAHE), which works based on a preset contrast limit (default is 40). If any histogram bin is above this contrast limit, the corresponding pixels are clipped and distributed equally to the remaining bins. Subsequently, histogram equalization is applied and any artifacts that might occur at the borders are removed through bilinear interpolation [34].

Fig. 10 shows the CLAHE results. Notice how the results are visually clearer, but the integrity of the image is still maintained. CLAHE has altered the histogram, but not completely flattened it; it has merely nudged the histogram in the right direction without going too far.

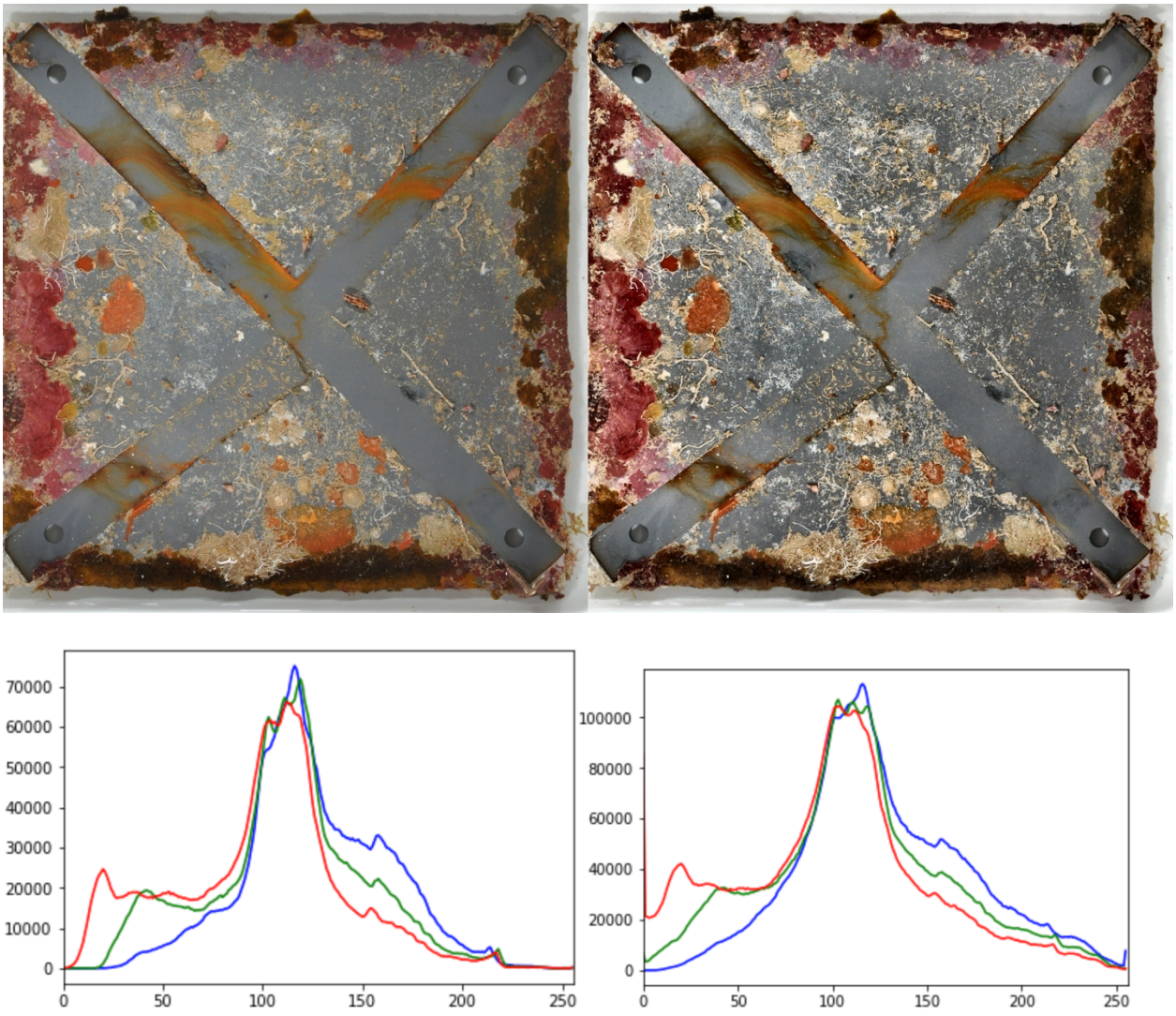

Fig. 10. (Left) Original image and its corresponding histogram. (Right) Image equalized based on contrast-limited locally adaptive histogram equalization, and the resulting histogram. The red, green, and blue lines correspond to the intensity distribution in the red, green, and blue image channels, respectively.

While CLAHE does deliver results that would aid researchers examining the data, it must still be ascertained to what extent CLAHE helps us with normalizing the images. An alternative approach could be to work in the HSV color space and apply either histogram equalization technique only on the V channel, leaving the hue and the saturation of the image untouched. These explorations are left to future work.

4.4 Cropping

It is worth noting that the plate images in this thesis have been cropped out from larger rectangular images to remove irrelevant visual information. This was achieved using a Python script prepared by the author of [27]. The script employs OpenCV's thresholding and contour finding functions to crop out the plate. Fig. 11 shows the uncropped version of the image discussed in Fig. 8. The extraneous details seen on the left and the right of the image would doubtless cause problems if not removed, and specifically, they interfere with histogram equalization.

Fig. 11. Uncropped version of the image discussed in Fig. 8.

On the other hand, it is possible that cropping out the plates could lead to some loss of information. Fig. 12 shows one such case, where significant growths are seen clinging to the edges of the plate. It might be important to devise an automatic way of checking for such potential data loss before cropping the images.

Fig. 12. Plate recovered off Rose Atoll, American Samoa, Pacific Ocean. The growths clinging to the side of the plates might be information researchers consider important. Measures might be needed to ensure they are not cropped out.

4.5 Advantages of the Proposed Approach

The proposed approach and the accompanying suggestions for future work in the next section offer several advantages. They

1. are easy to implement and experiment with
2. are computationally efficient compared to more sophisticated techniques
3. can be combined in various ways through bitwise operations to deliver a variety of useful results
4. can deliver quick results across an entire dataset if we figure out how to extract one particular kind of information, e.g., now that we have figured how to extract the colonies of red organisms, the method can be immediately applied to the entire dataset to quantify their population (provided we figure out image normalization)
5. will helps us eyeball the problem and come up with suitable evaluation criteria for future systems
6. might supplement future systems, e.g., histogram equalization would still be required to prepare data for artificial neural nets
7. can get us to a point where we have enough data for supervised-learning-based approaches

It is noteworthy that some of these techniques are reliant on ARMS standardization efforts, for example, having a uniform plate color and uniform perspective certainly helps these techniques work.

5. FUTURE WORK

The project carries tremendous possibilities for future work. These are outlined below along with useful references and suggested timescale:

1. Immediate:
    a. Test how well equalization works in terms of helping set uniform color and brightness ranges for all images. Experiment with working in HSV space and equalizing only the V channel (see Sec. 4.3).
    b. Once equalization works, test how well oxygen-producing red microorganism colonies can be quantified across a set of images by simply counting the extracted red pixels against total pixels.
    c. Extend the definition of "gray" to the brightness channel V too. This is crucial to generalizing the results for gray background removal. The relationship between saturation S and brightness V with respect to whether the color or hue H is discernible or not, is likely to be a non-linear function. The ranges will likely need to be discretized and the results combined with bitwise masks.
    d. Investigate whether region merging can be used for ARMS data. [35] is a popular textbook on computer vision that covers this topic, and it is available for free on the author's website.

2. Short-term:

    a. Investigate whether linear binary patterns (LBPs) can help us extract organisms exhibiting color patterns with some regularity (refer to [36] and [37]). The approach would be to first obtain texture datasets available online (e.g., the Brodatz Dataset [38] and the Kylberg Texture Dataset [39]. [40] provides an entire list at the bottom of the webpage). Obtain the LBPs for these patterns, then run them across images to check whether these patterns exist. The check will be done by monitoring some metric, e.g., Bhattacharya distance [28], and extracting the pattern if the distance is below a certain threshold.

Fig. 13. (Left) Texture of a scarf from the Kylberg Texture Dataset [39] and a shelled specimen with a similar pattern from the plate in Fig. 11.

    b. Test edge detection and contour detection on images in which certain colors are removed. Of course, removing a certain color could remove some part of a multicolored organism with a regular shape, breaking the contours. This could be addressed by removing one color at a time and checking for patterns, then two colors at a time, and so on. Since there are only about 6

main hues in the HSV space, the n is small enough to avoid scaling considerations (this is inspired by the recursive feature elimination approach used with SVMs [41]).

c. Devise an efficient and accurate scheme to extract unique images from the ARMS dataset, so that data is not over- or underrepresented. The dataset has several repeat photographs for each plate. However, there seems to be no way to determine from image metadata which group of photos correspond to a particular plate. The method used to address this would likely involve extraction of some descriptor from each image, which is then compared across images. One starting point could be to compute histograms for each image, use a suitable distance measure to compare them, and then cluster them using k-means or fuzzy c-means. Images of a given ARMS plate should be in one cluster.

3. Medium-term:

   a. Check whether the unsupervised segmentation technique in [26] is implementable, delivers good results, and is computationally feasible (see Sec. 2).

   b. If the above technique [26] can be realized, then a method will be required to perform unsupervised clustering. The iterative unsupervised clustering approach in [25] looks promising.

4. Long-term:
    a. A supervised deep learning implementation once enough data is available, supplemented with some of the techniques above, depending on need. It is important to note that even if there are some mistakes in the data, deep learning might be able to weed them out if the most incorrect or the most uncertain labels are examined (see Lesson 1 in [42]). Combining with cross-validation could help clean up the entire dataset [41].
    b. Make the DL implementation accessible over the web, with researchers having the option of contributing to an online database. This would further enable data collection and drastically accelerate research and conservation efforts.

6. CONCLUSION

This work explored the question of automatically extracting information from photographs of autonomous reef monitoring structures—standardized devices that can be deployed underwater for extended periods to collect samples of life from ocean habitats—in order to quantify the biodiversity of coral reef, the most sensitive and perhaps the most diverse of all marine habitats. Various computer vision (CV) techniques were analyzed to assess their suitability in the context of the problem—the complex (though in some ways standardized) nature of the visual data, the lack of labeled data to train the models, and the scalability and performance considerations for a task of this magnitude. Through a process of elimination, the study ended up exploring three simple image processing techniques: canny edge detection, color space transformations, and contrast-limited locally adaptive histogram equalization. While the canny edge detection results were not very encouraging, the other two turned out to be surprisingly efficient and effective in the context of the ARMS-CV project. The results led to the conclusion that such simple image processing techniques are the most promising avenue to get the project off the ground and collect enough data to train more sophisticated CV techniques (and to supplement them thereafter). To this end, concrete future directions of work were listed for the immediate, short, medium, and long term as a roadmap toward achieving the ultimate goal of creating an online automated analysis tool and marine life database that facilitates research and conservation efforts such as the Census of Coral Reef Ecosystems.

REFERENCES

[1] I. Lin *et al.*, "New evidence for enhanced ocean primary production triggered by tropical cyclone," *Geophys. Res. Lett.*, vol. 30, no. 13, 2003.

[2] R. E. Zeebe, J. C. Zachos, K. Caldeira, and T. Tyrrell, "Carbon Emissions and Acidification," *Science*, vol. 321, no. 5885, pp. 51–52, Jul. 2008.

[3] R. Costanza, "The ecological, economic, and social importance of the oceans," *Ecol. Econ.*, vol. 31, no. 2, pp. 199–213, Nov. 1999.

[4] R. Montaser and H. Luesch, "Marine natural products: a new wave of drugs?," *Future Med. Chem.*, vol. 3, no. 12, pp. 1475–1489, Sep. 2011.

[5] M. L. Reaka-Kudla, D. E. Wilson, and E. O. Wilson, *Biodiversity II: Understanding and Protecting Our Biological Resources*. Joseph Henry Press, 1996.

[6] J. B. C. Jackson, "Ecological extinction and evolution in the brave new ocean," *Proc. Natl. Acad. Sci.*, vol. 105, no. Supplement 1, pp. 11458–11465, Aug. 2008.

[7] K. E. Carpenter *et al.*, "One-third of reef-building corals face elevated extinction risk from climate change and local impacts," *Science*, vol. 321, no. 5888, pp. 560–563, Jul. 2008.

[8] O. Hoegh-Guldberg *et al.*, "Coral Reefs Under Rapid Climate Change and Ocean Acidification," *Science*, vol. 318, no. 5857, pp. 1737–1742, Dec. 2007.

[9] N. Knowlton, R. E. Brainard, R. Fisher, M. Moews, L. Plaisance, and M. J. Caley, "Coral Reef Biodiversity," in *Life in the World's Oceans: Diversity, Distribution, and Abundance*, A. McIntyre, Ed. John Wiley & Sons, 2010.

[10] A. Small, W. H. Adey, and D. Spoon, "Are current estimates of coral reef biodiversity too low? The view through the window of a microcosm," *Atoll Res. Bull.*, vol. 458, pp. 1–20, 1998.

[11] M. L. Reaka-Kudla, "Biodiversity of Caribbean Coral Reefs," in *Caribbean Marine Biodiversity*, P. Miloslavich and E. Klein, Eds. Lancaster, PA: Des Tech Publishers, pp. 259–276.

[12] "About the Census | Census of Marine Life." [Online]. Available: http://www.coml.org/about-census/. [Accessed: 16-May-2019].

[13] E. Ransome *et al.*, "The importance of standardization for biodiversity comparisons: A case study using autonomous reef monitoring structures (ARMS) and metabarcoding to measure cryptic diversity on Mo'orea coral reefs, French Polynesia," *PLOS ONE*, vol. 12, no. 4, p. e0175066, Apr. 2017.

[14] R. David *et al.*, "Lessons from photo analyses of Autonomous Reef Monitoring Structures as tools to detect (bio-)geographical, spatial, and environmental effects," *Mar. Pollut. Bull.*, vol. 141, pp. 420–429, Apr. 2019.

[15] Z. D. Stephens *et al.*, "Big Data: Astronomical or Genomical?," *PLOS Biol.*, vol. 13, no. 7, p. e1002195, Jul. 2015.

[16] T. Cordier, A. Lanzén, L. Apothéloz-Perret-Gentil, T. Stoeck, and J. Pawlowski, "Embracing Environmental Genomics and Machine Learning for Routine Biomonitoring," *Trends Microbiol.*, Dec. 2018.

[17] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning | Nature," *Nature*, vol. 521, May 2018.

[18] T. M. Press, "Deep Learning," *The MIT Press*. [Online]. Available: https://mitpress.mit.edu/books/deep-learning. [Accessed: 14-May-2019].

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *ArXiv151203385 Cs*, Dec. 2015.

[20] M. S. Norouzzadeh *et al.*, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proc. Natl. Acad. Sci.*, vol. 115, no. 25, pp. E5716–E5725, Jun. 2018.

[21] X. Sun *et al.*, "Transferring deep knowledge for object recognition in Low-quality underwater videos," *Neurocomputing*, vol. 275, pp. 897–908, Jan. 2018.

[22] A. Mahmood *et al.*, "Automatic annotation of coral reefs using deep learning," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–5.

[23] "ImageNet Large Scale Visual Recognition Competition (ILSVRC)." [Online]. Available: http://www.image-net.org/challenges/LSVRC/. [Accessed: 14-May-2019].

[24] "COCO - Common Objects in Context." [Online]. Available: http://cocodataset.org/#overview. [Accessed: 14-May-2019].

[25] J. Yang, D. Parikh, and D. Batra, "Joint Unsupervised Learning of Deep Representations and Image Clusters," *ArXiv160403628 Cs*, Apr. 2016.

[26] M. Kiechle, M. Storath, A. Weinmann, and M. Kleinsteuber, "Model-Based Learning of Local Image Features for Unsupervised Texture Segmentation," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1994–2007, Apr. 2018.

[27] K. Vojjila, "Image Segmentation and Classification of Marine Organisms," *Masters Proj.*, Apr. 2018.

[28] Yong Rui, A. C. She, and T. S. Huang, "Automated region segmentation using attraction-based grouping in spatial-color-texture space," in *Proceedings of 3rd IEEE International Conference on Image Processing*, 1996, vol. 1, pp. 53–56 vol.1.

[29] "About OpenCV." [Online]. Available: https://opencv.org/about/. [Accessed: 16-May-2019].

[30] "Just Color Picker," *Mac App Store*. [Online]. Available: https://itunes.apple.com/us/app/just-color-picker/id886547068?mt=12. [Accessed: 14-May-2019].

[31] E. L. Cavanaugh Nate, "Color Picker - AlloyUI." [Online]. Available: https://alloyui.com. [Accessed: 14-May-2019].

[32] "OpenCV: Canny Edge Detection." [Online]. Available: https://docs.opencv.org/master/da/d22/tutorial_py_canny.html. [Accessed: 17-May-2019].

[33] A. Rosebrock, "Zero-parameter, automatic Canny edge detection with Python and OpenCV," *PyImageSearch*, 06-Apr-2015. .

[34] "OpenCV: Histograms - 2: Histogram Equalization." [Online]. Available: https://docs.opencv.org/3.2.0/d5/daf/tutorial_py_histogram_equalization.html. [Accessed: 17-May-2019].

[35] "Computer Vision - Algorithms and Applications | Richard Szeliski | Springer." [Online]. Available: https://www.springer.com/us/book/9781848829343. [Accessed: 14-May-2019].

[36] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[37] A. Rosebrock, "Local Binary Patterns with Python & OpenCV," *PyImageSearch*, 07-Dec-2015.

[38] "Brodatz Textures." [Online]. Available: http://www.ux.uis.no/~tranden/brodatz.html. [Accessed: 17-May-2019].

[39] "Centre for Image Analysis." [Online]. Available: http://www.cb.uu.se/~gustaf/texture/. [Accessed: 15-May-2019].

[40] "Texture Synthesizability: predicting the quality of texture synthesis." [Online]. Available: http://www.vision.ee.ethz.ch/~daid/synthesizability/. [Accessed: 15-May-2019].

[41] "Introduction to Machine Learning with Applications in Information Security," *CRC Press*. [Online]. Available: https://www.crcpress.com/Introduction-to-Machine-Learning-with-Applications-in-Information-Security/Stamp/p/book/9781138626782. [Accessed: 17-May-2019].

[42] "Fast AI Deep Learning Course." [Online]. Available: https://course.fast.ai/videos/?lesson=1. [Accessed: 17-May-2019].