Coral Classification

Group 13

Chongyang Ben Du
c4du@ucsd.edu
Dept. of Computer Science and Engineering
UC San Diego

Shuyu Gu
sgu@ucsd.edu
Dept. of Computer Science and Engineering
UC San Diego

Mengyu Mo
m1mo@ucsd.edu
Dept. of Computer Science and Engineering
UC San Diego

Ziying Lisa Xue
zixue@ucsd.edu
Dept. of Computer Science and Engineering
UC San Diego

ABSTRACT

The coral reef ecosystems are diverse, fragile, and extremely important to the biosphere. Recently, they have been greatly impacted by the global climate change and have gained significant attention from research and government organizations. Our project intends to help the study on climate impact on different types coral reef colonies by providing a coral image classification methodology with machine learning models such as K-Nearest Neighbors, Random Forest, and Convolutional Neural Networks. These models we have chosen achieved test accuracy of 76.9%, 77.4%, and 65.8%, respectively. Source code has been made public on github: https://github.com/c4du/SIO_Coral_Classification

KEYWORDS

Coral, K-Nearest Neighbor, KNN, Random Forest, RF, Convolutional Neural Network, CNN

1 INTRODUCTION

Coral reefs have great importance to the marine ecosystems and benefit human cities in various ways. They provide shelter and habitat for marine organisms and protect coastline integrity from ocean waves and potential tropical storms. Different types corals have different metabolism and thus play different roles in forming reefs, such as barrier reefs and patch reefs. As humans continue to impact the climate, consequences like global warming and ocean acidification are causing increasing damage to the coral reefs, such as skeleton weakening and bleaching(aka coral death). To study such impacts on coral reef colonies as a whole, it is necessary to analyze how those climate change factors affect each type of coral individually. For example, certain types of coral which live farther away from the sea surface cannot perform photosynthesis due to insufficient sunlight, gain energy entirely from filtering algae and plankton, so they could be more susceptible to surrounding acidity changes[1, 2, 5, 11].

To help study and protect coral reefs, we are motivated to build a coral classification model to reduce the amount manual work necessary on classification, and also to further contribute to potential studies on coral bleaching patterns and coral colony health evaluation. Our classification method takes images of corals taken by underwater cameras as input, and outputs the predicted class label of the coral. The models we used are K-Nearest Neighbor, Random Forest, and Convolutional Neural Networks. For more detailed explanations see section 3 and 4.

2 RELATED WORK

Existing studies on coral image classification used several machine learning methods such as convolutional neural networks(CNN)[15, 16], support vector machine(SVM)[14, 17, 19], and K-nearest neighbor(KNN)[17, 19]. Ani Brown Mary et al. claimed that KNN models have better results on smaller datasets, but its performance on large datasets suffer from limited memory[17]. Amato et al. described the importance of the use of KNN models in classification problems that involve local feature similarities, we believe this technique could be useful in coral classification, since one would expect the same types corals have similar visual features [13]. The SVM technique was studied by our own UCSD computer vision group, which achieved the accuracy of 74.3% on coral image data within the year of 2008[14], we will not evaluate the SVM technique ourselves here. The current state-of-art method on coral image classification is indeed CNN and some of its variations such as DenseNet and Resnet. All studies using these techniques achieved accuracy over 95%, and we will attempt to repeat the experiments.

We realized that one popular machine learning model was missing from the set of papers we have read - random forest. In this paper, we will build a random forest model, evaluate its performance, and compare with the others.

3 DATASET AND FEATURES

We choose the Moorea Coral Reef LTER dataset, which contains over 2,000 coral reef images from the island of Moorea in French Polynesia from 2008 to 2010. Each coral image has 200 random points annotated with one of the 9 classes for which 5 of them are coral genera (Acropora, Pavona, Montipora, Pocillopora, and Porites) and 4 are non-coral classes (Crustose Coralline Algae, Turf algae, Macroalgae and Sand)[4]. In the original dataset, each picture contains multiple corals of different classes. For simplicity, we will use a processed dataset from the UCSD Computer Vision group, which contains 32,686 cropped images such that each image contains only one class of coral/algae[12]. The number images of each class of coral/non-coral is shown in the figure below.

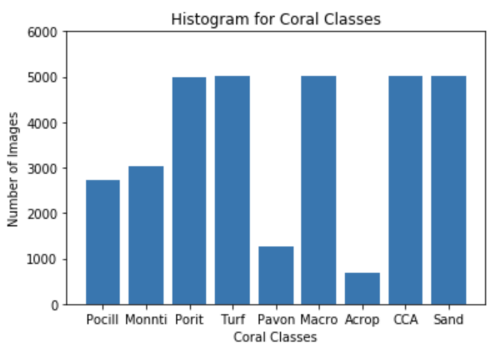


Figure 1: Number of Images of Corals of each Class

To train the model, we are going to extract several features from these images The extracted local features must be:

- Repeatable and precise so they can be extracted from different images showing the same object.
- Distinctive to the image, so images with different structure will not have them.

There is a wealth of algorithms satisfying the above requirements for feature detection (finding interest points on an image) and description (generating a vector representation for them). They include Harris Corner Detection, Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Features from Accelerated Segment Test (FAST), and Binary Robust Independent Elementary Features (BRIEF), keypoint detector and descriptor extractor (KAZE). For this project, we will use KAZE for feature detection and description, it's part of the opencv library. We will discuss examples in the section below.

4 METHODS

Image feature is a simple image pattern, based on which we can describe what we see on the image. For example cat eye will be a feature on a image of a cat. The main role of features in computer vision(and not only) is to transform visual information into the vector space. This give us possibility to perform mathematical operations on them, for example finding similar vector(which lead us to similar image or object on the image), train different models based on the description vector. There are two ways of getting features from image, first is an image descriptors, second is a neural network.

4.1 Feature Extraction

4.1.1 RGB value matrices (raw image). We got the RGB matrix corresponding to each image. For example, 0-255 are the possible integer values, 0 is minimum for any RGB color, 255 is maximum for any RGB color. To define any color, you must specify the Red(R), Green(G) and Blue(B) values.

4.1.2 Gray scale images. Color images are often built of several stacked color channels, each of them representing value levels of the given channel. For example, RGB images are composed of three independent channels for red, green and blue primary color components; CMYK images have four channels for cyan, magenta, yellow and black ink plates, etc. Each channel is represented using a gray level image. Since For many applications of image processing, color information doesn't help us identify important edges or other features. Moreover, color increases the complexity of the model. For contrast, to see whether color information is useful, we also convert images into grayscale mode.

4.1.3 KAZE descriptor with mode RGB. With KAZE descriptor, we first detect keypoints on image(center points of our local patterns). The number of them can be different depend on image so we add some clause to make our feature vector always same size(this is needed for calculation, cause you can't compare vectors of different dimensions) Then we build vector descriptors based on our keypoints, each descriptor has size 64 and we have 32 such, so our feature vector is 2048 dimension. For example, the extracted features for two coral images from same class is described in figure 2, and the top 10 matched features for above two images are described in figure 3.

4.1.4 HSV (Hue, Saturation, Value) histogram. Consider a histogram as a graph or plot, which gives you an overall idea about the intensity distribution of an image. It is a plot with Saturation (0-255) on X-axis and Hue (0-180) on Y-axis. From the properties of the HSV color space, it is observed that for low values of saturation, a color is approximated by a gray value specified by the intensity level while for higher saturation, the color is approximated by its hue. An example of original and HSV-converted image of a cauliflower coral is shown in figure 4 and its HSV histogram in figure 5.

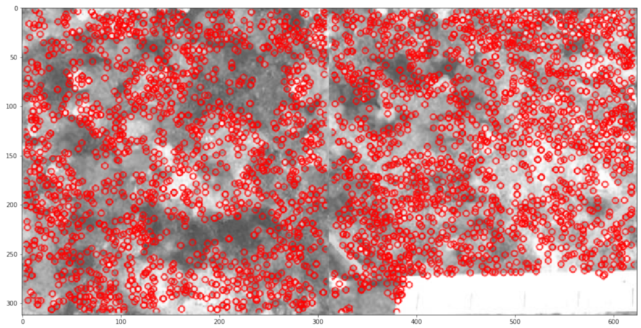


Figure 2: Extracted features for two coral images from same class

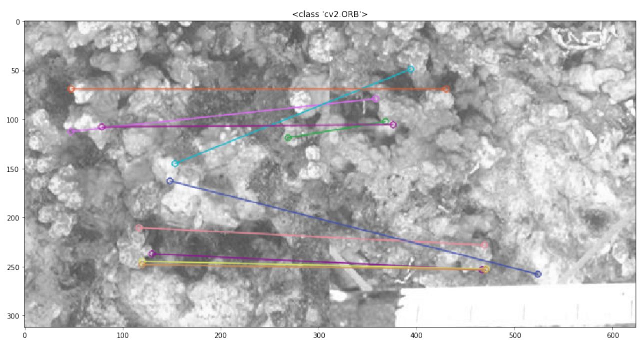
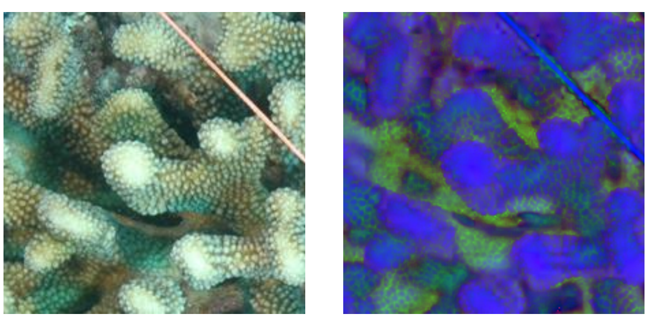


Figure 3: Top 10 matched features



(a) RGB Image **(b) HSV Image**

Figure 4: Original and Processed Image of Pocillopora (Cauliflower Coral).

4.2 Training Models

4.2.1 K Nearest Neighbors. K Nearest Neighbour Classifier is used to classify objects with the labels of highest frequency from its top K similar neighbors. We apply the sklearn neighbor module[7] which implements the Nearest Neighbor Classifier on our pre-processed data and pick different K to examine the train error and test error. It turned out that K = 2 achieves the best results on our dataset.

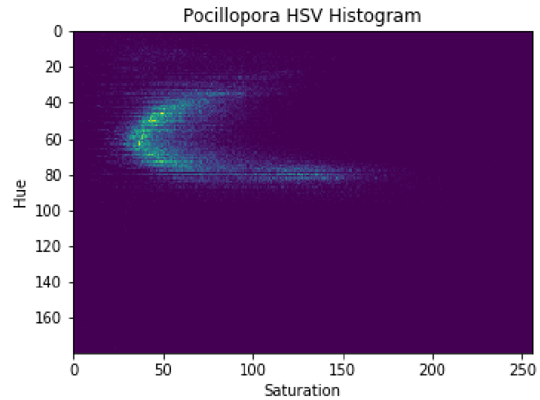


Figure 5: HSV Histogram of Image from Figure 4. Most common Hue and Saturation are around 60 and 40 respectively.

4.2.2 Random Forest. Random Forest Classifier applies a number of decision tree by choice on the dataset and uses averaging to improve the predictive accuracy and control over-fitting[9]. We apply the sklearn ensemble module which implements the Random Forest Classifier on pre-processed data. After trying different n_estimators and tuning the max_depth of trees, examining the train and test error, we found out 200 trees with maximum depth 20 give the best performance.

4.2.3 Convolutional Neural Network. Instead of extracting features first and then passing specific features into the models as the input, we have also used Convolutional Neural Network(CNN) to classify coral images. The most important layers of a CNN model is Convolution, Pooling and Flattening. A fully connected CNN model can have multiple layers for each. Convolution layer mainly serves to extract features from the images. What's good about convolution layer is that it still preserve the spatial relationship between pixels. In our model, we set kernel size(the size of the matrix to be convoluted with) to be 5x5. Since our image is of size 221x221, the image matrix resulted from convolution would shrink down to a size of 217x217. And the first layer contains 32 of such filters to get us a number of feature maps derived from a single coral image. We have also applied an activation layer right after convolution using "ReLU" in order to introduce non-linearity.
The second significant layer is Pooling Layer. The main goal of Pooling is to reduce the dimensionality of the feature maps fed from convolution layer, while keeping the most important pixel information. In our model, we decided to use MaxPooling algorithm, in which we extract the pixel with the maximum value from a neighbor of size 2x2. Now we get

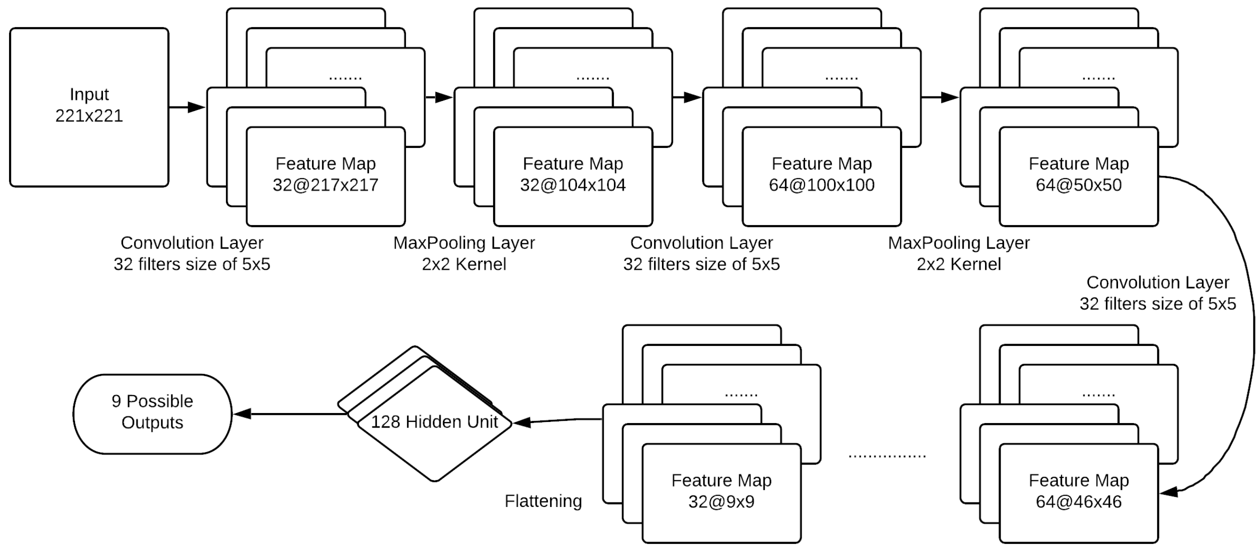


Figure 6: CNN Layering Architecture

a number of rectified feature map. It is definitely smaller in size, but we still keep largest amount of information from the original image. In order to find the balance between training and testing and avoid overfitting on training set, the dropout layer in the model is set to 0.7, which means that only a random 30% of the neurons would be awake during training phase. Flattening Layer at last converts the matrix into a linear array to be passed to the nodes in the model. At last when connecting all layers, we use "softmax" activation function in order to classify images into 9 different labels. The detailed CNN architecture is described in Figure 6. The implementation of CNN was done with the help of the documentation and examples from Tensorflow and Keras[3, 10].

5 EXPERIMENTAL AND EVALUATION

Our dataset contains 32,686 images in total. We split them into training set and testing set by percentage 75% and 25%. CNN are trained using 80% training and 20% testing. We first use accuracy/error rate to evaluate our three models, and then we further evaluate our KNN and Random Forest model using precision and recall. Cross validation was not automated in this project, as we manually tried several values for the parameters for the models, and we concluded the following parameters were optimal for each model:

- KNN: n_neighbors= 2, algorithm = kd-tree
- Random Forest: n_estimators = 200, max_depth = 20

All the evaluation results of KNN and random forest models listed below used HSV histogram as feature input, which produced the best results. We have attempted to raw RGB value matrices as feature input, and decided that this choice of feature is inefficient because training time could range from 40 minutes to 1 hour. The other feature choices are further discussed in section 6.

Table 1 shows the error rates of the three models we ran, and random forest achieved the highest accuracy on both training set and testing set. Table 2. Figure 7 contains the confusion matrices for KNN and Random Forest models, which was drawn by the sklearn confusion matrix library and seaborn heatmap library[6, 8]. Since there are very few Acrop coral images in our dataset, the very top left grid does not show a outstanding number for correct classification of Acrop. for Some observations include: CCA is one of the most commonly misclassified labels for both models.

Model	Train Error	Test Error
RF	0.2%	22.2 %
KNN	9.5%	23.1%
CNN	10.2%	34.2 %

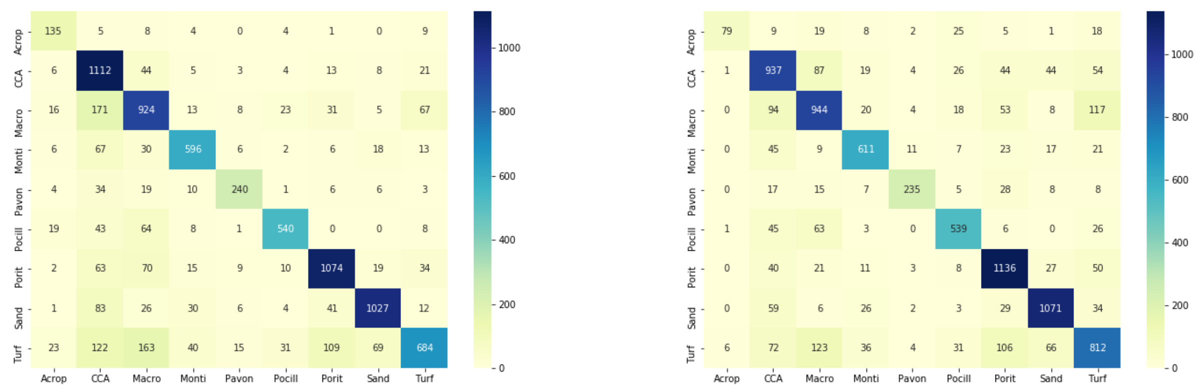
Table 1: Train and Test Error of different models.

6 DISCUSSION

For Random Forest Classifier, we first train the regressor with extracted features by using KAZE descriptor, then, we got 47.8% error rate for test data set. The error rate is little bit higher than we expect, similar images are not always

Coral/Non-coral type	KNN		Random Forest	
	Precision	Recall	Precision	Recall
Acrop	0.64	0.81	0.91	0.48
CCA	0.65	0.91	0.71	0.77
Macro	0.69	0.73	0.73	0.75
Monti	0.83	0.80	0.82	0.82
Pavon	0.83	0.74	0.89	0.73
Pocill	0.87	0.79	0.81	0.79
Porit	0.84	0.83	0.79	0.88
Sand	0.89	0.83	0.86	0.87
Turf	0.80	0.54	0.71	0.65

Table 2: Classification Report



(a) KNN **(b) Random Forest**

Figure 7: Confusion Matrix Evaluation on Classification Models

similar as we understand it. After we did some research, we found that this algorithms is context-unaware, so they better in finding same images even modified, but not similar. Therefore, we decided to use HSV histogram and RGB value matrices to train this model, you can consider histogram as a graph or plot. In this way, we got 22.2% error rate for test data set. Therefore, we can conclude from above that the HSV histogram is a good way to represent these coral images.

We have encountered some difficulties while training our convolutional neural network since we expected to see higher accuracy by using CNNs. Common architectures such as Conv-Pool-Conv-Pool and Conv-Conv-Pool-Conv-Conv-Pool, even after several tries of fine-tuning hyper parameters, did not produce a reasonable testing accuracy, 10% while achieving very high training accuracy 90%. After re-examining the dataset, we found that most coral images are dim and blurry (with weird lightings) which made us believe that bigger amount of pixels might be needed for network to recognize the coral (Kernel size increased from (3,3) to (9,9)). Based on this, we manually added more Convolution Layers(with increasing number of filters, larger filter size) and performed normalization. Dropout layers with different values(0.3-0.7) were also added to avoid overfitting. Eventually we managed to bring testing accuracy to 65.8% with training accuracy to be 89.9%.

7 CONCLUSION

In this project, we built, evaluated, and compared 3 coral image classification models with 4 methods of image feature extraction. The 4 feature extraction methods we compared are RGB value matrix, gray scaling, KAZE descriptor, and HSV histogram. After evaluating their runtime and accuracy, we concluded that HSV histogram is the optimal feature extraction method. The 3 classification methods we explored are K-Nearest Neighbor, Random Forest, and Convolutional Neural Network, among which Random Forest achieved the highest accuracy on both training set and testing set. The test accuracy of the above three models are 76.9%, 77.8%, and 65.8%, respectively. Although we were not able to reproduce the highly accurate results of the state-of-art CNN models, we believe that this project contributed to the field of coral classification by exploring the seldom used Random Forest model. This project also successfully served its pedagogical purpose, in which we experienced the practical process of build machine learning models, parameter tuning, and layering design in neural networks.

8 LIMITATIONS AND FUTURE WORK

Due to the time limitation of the course and insufficient background marine expertise on corals, we only got the chance to apply four feature types of extraction methods and three different models on our dataset and were not able to reproduce the CNN models with high accuracy mentioned in section 2. For future studying, We could improve our research by gaining broader knowledge on corals and its dependent marine organisms, obtaining more comprehensive understanding of different machine learning models, and exploring different feature extraction and augmentation methods. For example, we can try to improve image pre-processing techniques by trying different patch sizes, as ambiguous features may be removed. For our HSV histogram feature extraction method, we could try image augmentation and enhancement, such as amplifying the hue or saturation for all coral images. We also would apply more machine learning models suitable for image classification such as ResNet, NASNet, and tuned more parameters to get more desired results.

9 CONTRIBUTION

Each group member has been of equal importance in the completion of this project, and all members were involved in all parts of the project. However, each team member did have a different focus on this project. Chongyang Ben Du contributed mainly to researching background information about marine corals by discussing project ideas with the professor and marine scientists. Mengyu Mo primarily chose the method for pre-processing the data and apply four different kinds of feature extraction methods. Shuyu Gu mainly build and tuned the K Nearest Neighbor and Random Forest models by looking up documentations and laboriously repeating experiments, and Ziying Lisa Xue significantly contributed to designing the layers in the Convolutional Neural Network model.

REFERENCES

[1] 2019. Corals and Coral Reefs. https://ocean.si.edu/ocean-life/invertebrates/corals-and-coral-reefs

[2] 2019. Human Impact on The Reef. https://www.qm.qld.gov.au/microsites/biodiscovery/05human-impact/importance-of-coral-reefs.html

[3] 2019. Keras Documentation MNIST CNN. https://keras.io/examples/mnist_cnn/

[4] 2019. Moorea Coral Reef LTER. http://mcr.lternet.edu/cgi-bin/showDataset.cgi?docid=knb-lter-mcr.5006

[5] 2019. Scientists Pinpoint How Ocean Acidification Weakens Coral Skeletons. https://www.whoi.edu/press-room/news-release/scientists-identify-how-ocean-acidification-weakens-coral-skeletons/

[6] 2019. Scikit-learn Confusion Matrix. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

[7] 2019. Scikit-learn K-Neighbors Classifier. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[8] 2019. Seaborn Heatmap. https://seaborn.pydata.org/generated/seaborn.heatmap.html

[9] 2019. Sklearn Ensemble Random Forest Classifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[10] 2019. Tensorflow Advanced CNN. https://www.tensorflow.org/tutorials/images/deep_cnn

[11] 2019. Types of Coral Reef Formations. https://coral.org/coral-reefs-101/coral-reef-ecology/types-of-coral-reef-formations/

[12] 2019. UCSD Computer Vision Moorea Labaled Corels. http://vision.ucsd.edu/content/moorea-labeled-corals

[13] Giuseppe Amato and Fabrizio Falchi. 2010. KNN based image classification relying on local feature similarity. *Proceedings - 3rd International Conference on SImilarity Search and APplications, SISAP 2010*, 101–108. https://doi.org/10.1145/1862344.1862360

[14] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman. 2012. Automated annotation of coral reef survey images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 1170–1177. https://doi.org/10.1109/CVPR.2012.6247798

[15] Anabel GÃşmez-RÃŋos, Siham Tabik, JuliÃąn Luengo, ASM Shihavuddin, Bartosz Krawczyk, and Francisco Herrera. 2019. Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation. *Expert Systems with Applications* 118 (2019), 315 – 328. https://doi.org/10.1016/j.eswa.2018.10.010

[16] Ammar Mahmood, Mohammed Bennamoun, Ferdous Sohel, Senjian An, Farid Boussaid, Renae Hovey, Gary Kendrick, and Robert Fisher. 2017. *Deep Learning for Coral Classification*. https://doi.org/10.1016/b978-0-12-811318-9.00021-1

[17] N. Ani Brown Mary and Dejey Dharma. 2017. Coral reef image classification employing Improved LDP for feature extraction. *Journal of Visual Communication and Image Representation* 49 (2017), 225 – 242. https://doi.org/10.1016/j.jvcir.2017.09.008

[18] Andrey Nikishaev. 2018. Feature extraction and similar image search with OpenCV for newbies. *Medium* (2018). https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774

[19] A.S.M. Shihavuddin, Nuno Gracias, Rafael Garcia, Arthur C. R. Gleason, and Brooke Gintert. 2013. Image-Based Coral Reef Classification and Thematic Mapping. *Remote Sensing* 5, 4 (2013), 1809–1841. http://www.mdpi.com/2072-4292/5/4/1809