MDNet: Multi-Patch Dense Network for Coral Classification

Md Modasshir
University Of South Carolina
modasshm@email.sc.edu

Alberto Quattrini Li
Dartmouth College
Alberto.Quattrini.Li@dartmouth.edu

Ioannis Rekleitis
University Of South Carolina
yiannisr@cse.sc.edu

***Abstract*—Classifying coral species from visual data is a challenging task due to significant intra-species variation, high inter-species similarity, inconsistent underwater image clarity, and high dataset imbalance. In addition, point annotation, the labeling method used for coral reef images by marine biologists, is prone to mislabeling. Point annotation also makes existing datasets incompatible with state-of-the-art classification methods which use the bounding box annotation technique. In this paper, we present a novel end-to-end Convolutional Neural Network (CNN) architecture, Multi-Patch Dense Network (MDNet) that can learn to classify coral species from point annotated visual data. The proposed approach utilizes patches of different scale centered on point annotated objects. Furthermore, MDNet utilizes dense connectivity among layers to reduce over-fitting on imbalanced datasets. Experimental results on the Moorea Labeled Coral (MLC) benchmark dataset are presented. The proposed MDNet achieves higher accuracy and average class precision than the state-of-the-art approaches.**

***Index Terms*—Deep learning, Convolutional neural networks (CNN), Marine images, Classification, Marine ecosystems**

I. Introduction

Coral reef ecosystems are very important for a number of reasons: they are hosts to many species and they play an important role in the capture of CO_2 in the water. Unfortunately, coral reef populations are on a rapid decline [1]. Thus, monitoring the health of coral ecosystems through scuba divers and new technologies, such as underwater robots, has become more and more important, resulting in the acquisition of millions of images. While image acquisition has evolved, reef monitoring still requires the identification of the different coral species, a task that is mainly performed by human experts.

The automated detection of coral species in a video feed is a challenging problem, due to the underwater environment that causes, among others, hazing, blurring, light variation/absorption. In addition, coral species exhibit high intra-class and inter-class variability. Besides, coral reef are densely populated resulting in complex spatial borders among the different classes. As a consequence, human experts often struggle to analyze images and to accurately identify different coral species [3]. Another challenge is the presence of numerous dead corals on which different species of algae grows. These algae together with dead coral exhibit similar shape and texture to the live specimens [4] of coral species. As such, the classification becomes harder using traditional handcrafted features.

In recent years, object classification using Convolutional Neural Network (CNN) has become a great success story. The introduction of several new techniques using CNN such as AlexNet [5], VGGNet [6], ResNet [7], and DenseNet [8] has resulted in recognition rates surpassing human level accuracy in popular benchmark datasets, e.g., [9], [10]. Although deep learning architectures have shown superior performance for visual recognition activities, there is limited work in the area of coral classification.

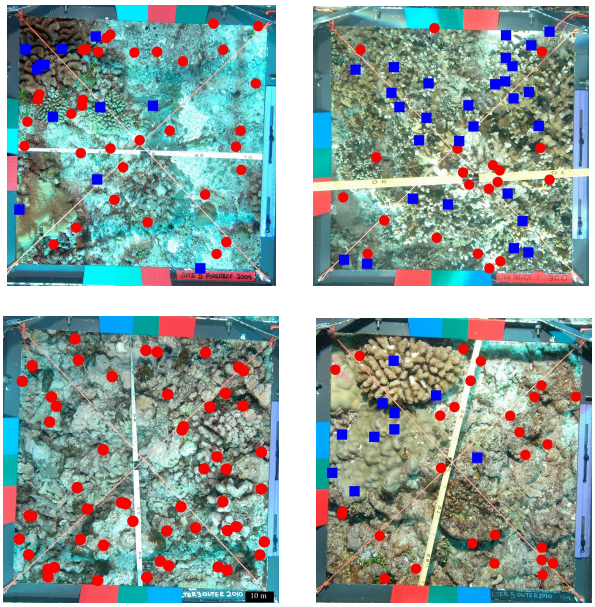

Fig. 1: Sample images from MLC dataset [2] with annotated points. Blue rectangles represent coral classes and red circles indicate non-coral classes.

For solving the classification problem, traditional deep neural networks require either image level annotation or bounding box annotation. The MLC benchmark dataset [2] uses point annotation. In the point annotation method, a number of points are randomly sampled in the image and experts are asked for each point whether the point is on a coral or not. Some examples of point annotation on the MLC dataset is shown in Fig. 1. This annotation method is popular among marine biologists. Mahmood et al. [11] employing transfer learning of CNN creates fixed sized patches centered on the labeled point and resizes the cropped images to be used as input to the proposed network. Resizing each patch, however, is an expensive operation and causes loss of information. For the MLC dataset, the cropped patches often include only a small portion of the object it is labeled for—see Fig. 2g—while sometimes cropping a smaller size patch (28×28) does not

978-1-5386-4814-8/18/$31.00 ©2018 IEEE

(a) (b) (c) (d) (e) (f) (g)

Fig. 2: (a) Example of point annotated image shows some points not on coral; (b) & (c) Pocillopora patches of size 28×28 and 56×56. For the Pocillopora patches, the texture and shape information is available at both crop sizes. However, for Montipora patches, smaller patches contain no information about the coral object. Hence bigger patch size is necessary in such cases. (d) & (e) & (f) & (g) Montipora patches of size 28×28, 56×56, 84×84, and 112×112, respectively.

include the coral at all as shown in Fig. 2d. Therefore, finding the appropriate crop size is challenging.

A major bottleneck on coral classification is the imbalance of the number of samples for classes of different coral species—e.g., in the MLC dataset [2], Crustose Coralline Algae (CCA) 48% vs. Acropora 0.01%. This inherent class imbalance in coral datasets makes the CNN training process complicated. Transfer learning has shown promising result in handling imbalanced datasets in other domains. Transfer learning refers to the approach where the representation from state-of-the-art networks pre-trained on a large dataset (e.g., ImageNet [12]) is extracted and then fed to a classifier. Another approach is balancing the target dataset by augmenting less frequent classes or sampling equal number of samples per class. After balancing the dataset, a CNN can be trained from scratch provided that enough samples are present compared to the number of the hyper-parameters of the network. In practice, for small but balanced datasets, fine-tuning large networks (e.g., VGG [6]) provides excellent performance. Fine-tuning is performed by disabling hyper-parameters update for most of the initial layers of a pre-trained network and only training few final layers.

In our proposed method, the Multipatch Dense Network (MDNet) fits annotated points in parallel on patches of several sizes (as big as 84×84 or even 112×112) and then aggregates the learned representation to classify the image with the appropriate coral species. Therefore, the network is able to use information from different scales. Moreover, to prevent over-fitting, the network is designed utilizing dense connectivity between layers. In addition, we train the network with a cost sensitive loss function to favor infrequent classes. Our proposed network performs better on the benchmark MLC dataset than state-of-the-art algorithms in terms of accuracy and average class precision. To summarize, our **primary contributions** are:

1) An end-to-end deep learning framework, MDNet, to classify coral species.
2) The use of multi-scale patches from point annotated visual data to train a deep learning model.

II. Related Work

Most of the traditional approaches for coral classification focused on pixel-based classifiers where the spectral properties of each pixel of the image were used to determine each coral's class. Textural appearances were used by Mehta et al. [13] to classify coral reef images. The approach used in [13] could not handle illumination changes in the underwater environment. Based on image color and texture features extracted from the coral reef video frames, Marcos et al. [14] developed an automated rapid reef classification system, able to perform finer scale data acquisition and processing compared to previously existing systems. Marcos et al. [14] adopted the histogram of normalized chromaticity coordinates (NCC) to extract color features, and the local binary patterns (LBP) descriptor to extract texture features from the coral reef images. Stokes et al. [15] proposed a coral classification method based on the discrete cosine transform and a k-nearest neighbor classifier. Even though the results appeared promising, only a small dataset, containing just 16 images, was used. It is not clear whether the approach proposed in [15] would generalize well. Early work by Beijbom et al. [2] proposed a multiple scale classification algorithm using texture and color descriptors. The MLC dataset with 400,000 expert annotations was introduced. Their approach is the first one that addressed automated annotations of coral reef survey image on a large scale. The Maximum Response (MR) filter bank [2] was used for texture and color feature extraction. The filter was applied to each color channel in the $L \times a \times b$ color space and then the filter response vectors were stacked. Beijbom et al. [2] used Support Vector Machines with Radial Basis Function kernel as classifier.

Mahmood et al. [11] proposed a method using learned features and hand-crafted features from multi-scale patches. First, four patches centered on the annotated points were extracted and resized to the input size of VGGnet [6]. Then, feature representations were extracted from the last layer of VGGnet (16 layer configuration) before the first fully connected layer. At this point, hand-crafted features were extracted and concatenated with extracted features from VGGnet. Finally, these concatenated features are fed to a 2-layer MLP (multilayer perceptron) classifier. While our proposed network also utilizes several patch sizes, our model learns from several patches with no resizing operations and aggregates the learned features to produce a prediction. Moreover, our approach is an end-to-end model while the approach proposed by Mahmood et al.

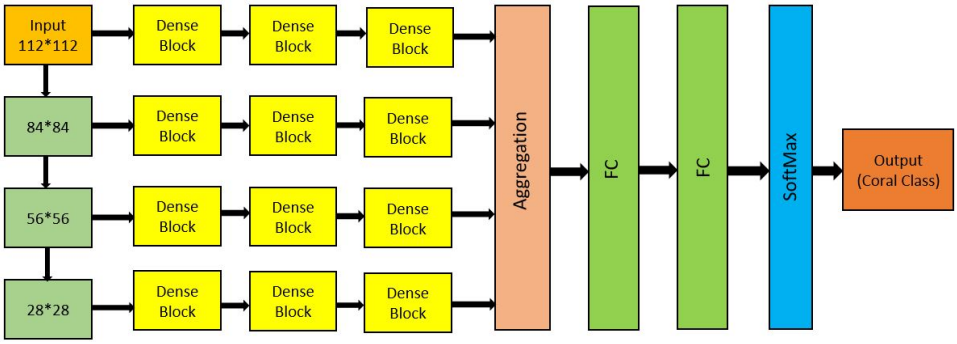

Fig. 3: Block diagram of the proposed coral classification framework.

[11] is a multi-step process. Class dependent costs are used to calculate the loss of the network to reduce the effect of class imbalance.

More recently, DenseNet [8] has shown promising result in the classification of popular object (non-coral) datasets such as CIFAR-10, CIFAR-100, SVHN, and ImageNet using a smaller number of hyper-parameters. Huang et al. [8] showed that dense connections between layers allow the network to learn faster because of the improved flow of information and the faster propagation of gradients during back-propagation. Dense connections refer to the fact that each layer is connected to all subsequent layers up to the loss layer. Because of this connection between the loss layer and each previous layer, the network learns in a deeply supervised manner, which helps reducing over-fitting in smaller size training sets.

III. Proposed Method

We propose a novel deep learning framework, *Multi-Patch Dense Network* (MDNet), for coral classification inspired by DenseNet [8] architecture. The benchmark dataset MLC follows point annotation technique. The point annotation technique suggests that the annotated point is located on an object; thus, the comparatively smaller patches (28×28) containing the annotated points in the center will best reflect the object. In practice, this idea in general holds true for most smaller patches. However, there is a number of annotated points that are not exactly on the object; see Fig. 2. Surprisingly, there even smaller patches where no object is present. On the other hand, larger patches (112×112 or 84×84) always contain the object it is annotated for. However, larger patches, in many cases, may contain several different species of coral and parts of the background (non-coral). In our empirical analysis, we observed complex spatial boundary among corals species, which is why, it is nearly impossible to find a patch size that will contain only the object, applicable for all points. Thus, it is imperative to consider both smaller and larger patches to extract useful information such as texture and shape.

The key idea proposed in this paper is to start with a larger patch (112×112) and iteratively crop towards the center. As a result, the network can utilize the varying patch sizes to learn the texture and appearance of distinct coral species. Another important feature of the network is that it is densely connected among layers to reduce over-fitting.

A. Network Architecture

Our network architecture consists of several building blocks inspired by DenseNet. We first explain these building blocks and then present our network architecture.

1) Dense Layer: Consider a single patch x_0. We define a composite function dense layer, D_l where l denotes the layer. D_l is a combination of three standard CNN operations in sequence: batch normalization (BN) [16], rectified linear unit (ReLU) [17], and a 3×3 convolution. The output of the l-th dense layer is denoted by x_l.

2) Transition Layer: The transition layer consists of a 1×1 convolution followed by a 2×2 average pooling with stride 2. The transition layer is placed between dense blocks.

3) Dense Block: Several dense layers are connected sequentially to create a dense block. Each of the dense layers in the dense block is connected to all subsequent dense layers in the same block in a feed-forward fashion, as shown in Fig. 4. For the l-th dense layer, there are l inputs consisted of the feature maps of all preceding layers, $[x_0, ..., x_{l-1}]$. Here $[x_0, ..., x_{l-1}]$ indicates the concatenated feature maps produced in layer $0, ..., l-1$. Therefore, the input of the l-th layer is:

$$x_l = D_l([x_0, x_1, ..., x_{l-1}]) \quad (1)$$

Within a dense block, there is no pooling layer. Each layer has input from all the preceding layers within the same dense block, so the number of feature maps grows fast. This proliferation of feature maps may exceed the limit of GPU memory. Therefore, our network is designed with a smaller number of dense layers per block.

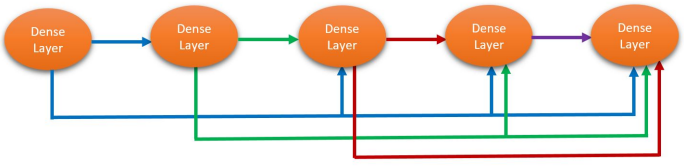

Fig. 4: Illustration of a dense block.

4) CNN Architecture: MDNet is shown in Fig. 3. We define a dense pipeline as a sequence of 3 dense blocks connected by transition layers. The input size of our proposed network is 112×112. The input image, x_0, is fed to a dense pipeline. In parallel, the input is connected to center crop layer, $crop_1$ which crops the input images to 84×84 size. Another dense pipeline takes the output of the $crop_1$ layer. Again, $crop_1$ layer is followed by another center crop layer $crop_2$ and a dense pipeline. The $crop_2$ layer extracts a 56×56 size patch from the center of the input. In parallel, $crop_2$ is connected to a center crop layer $crop_3$ which extracts a 28×28 patch. The $crop_3$ layer is again connected to a dense pipeline. At this point, the outputs of the four parallel dense pipelines are converted to a 1-D vector and concatenated. These concatenated features are fed to two fully connected layers with 2048 hidden units followed by a softmax layer.

To measure the loss of the proposed model, we have used cross-entropy. The softmax layer takes the learned representation, f_{i} and interprets it to the output class. A probability score p_{i} is also assigned for the output class. If we define the number of coral classes as K, then we get

$$p_i = \frac{\exp(f_i)}{\sum_i \exp(f_i)}, i = 1, ..., K \tag{2}$$

and

$$L = -\sum_i g_i \log(p_i) \tag{3}$$

where L is the loss of cross entropy of the network. Back propagation is used to calculate the gradients of the network. If the ground truth of an input image is denoted as g_{i}, then,

$$\frac{\partial L}{\partial f_i} = p_i - g_i \tag{4}$$

5) Cost Sensitive Training: To tackle the class imbalance problem, we use cost sensitive training [18]. The output of the softmax layer, p_i is modified by a cost matrix C to calculate class dependent costs. Therefore, the modified class dependent output y_n of the network is as follows:

$$y_n = \frac{C_{p,n} \exp(f_i)}{\sum_i C_{p,n} \exp(f_i)} \tag{5}$$

where p is the desired class and N is the number of neurons in the output layer. The modified output of the network, y_n, is then used to calculate the class dependent loss:

$$L = -\sum_i g_i \log(y_n) \tag{6}$$

The cost function only affects the loss calculation, and does not change the gradients in back-propagation [18]. The diagonal values of the cost weight matrix, C, represent the weight for each class of K. The weight of a class k_i is calculated as:

$$k_i = 1 - \frac{S_{k_i}}{\sum_{k=1} S_{k_i}} \tag{7}$$

where S_{k_i} is the number of samples in class k_i.

B. Model Training

1) Preprocessing: To address the challenge of point annotated data, the following preprocessing steps are followed. A patch of size 112×112 is extracted for every annotation point with the annotated point in the center of the patch. We perform normalization on the patches before feeding to the network. Normalization changes the range of pixel intensity values and transforms an n-dimensional image $I : \{\mathbb{X} \subseteq \mathbb{R}^n\} \rightarrow \{X_{\min}, ..., X_{\max}\}$ with intensity values in the range $(X_{\min}, X_{\max})$ into a new image $I_N : \{\mathbb{X} \subseteq \mathbb{R}^n\} \rightarrow \{Y_{\min}, ..., Y_{\max}\}$ with intensity values in the range $(Y_{\min}, Y_{\max})$. The linear normalization of a gray-scale digital image is performed according to the formula:

$$I_N = (I - X_{\min}) \frac{Y_{\max} - Y_{\min}}{X_{\max} - X_{\min}} + Y_{\min} \tag{8}$$

2) Augmentation: For the MLC dataset, all the images are collected using the same protocol: using a frame over coral reef. Therefore, we only choose to augment these datasets using random horizontal flip, random image channel shift, and random rotation between 20° and -20°. Random image channel shift is chosen to make the network robust to color suppression since red color is the first color absorbed in underwater images.

3) Training Hardware and Parameters: We train the model on two Nvidia P100 GPUs with a mini-batch of 512. We implemented our network using Keras with Tensorflow backend. After some experiments, we found the following parameters to work best for our model: learning rate 0.01, weight decay $1e-4$ every 100 iterations and Nesterov momentum 0.9.

IV. Experiments & Results

MDNet has been tested on a standard benchmark dataset for coral species classification, **MLC** Moorea Labeled Corals (MLC) dataset [2]. MLC dataset [2] consists of a subset of images, collected at the Moorea Coral Reef Long Term Ecological Research site (MCR-LTER). The images are collected from three different habitats over a three year period (2008-2010). Images of each particular year are considered as a separate dataset. In total, there are 2055 images that have a large variety in coral shape, color, scale, and viewing angle. For our experiments, images of the nine most frequent classes of the MLC dataset were used; where five are corals and four are non-corals.

A. Experimental Setup

For the MLC dataset, we set up our experiments in a similar way to Beijbom et al. [2]. In the first set up—Experiment 1—we randomly split data collected on 2008 into 80/20 train/test set. In the second set up, Experiment 2, we train our model on 2008 transect and test on 2009 transect. In the third set up, Experiment 3, we train on 2008 and 2009 transects combined, and test on 2010 transect. We report the performance of the models in terms of accuracy and average class precision [19].

Baseline: We consider the work by Mahmood et al. [11] as baseline and we call their method CF. As there is no

Model	Accuracy(%)			ACP(%)		
	Exp1	Exp2	Exp3	Exp 1	Exp 2	Exp 3
CF	77.9	70.1	84.5	69	63	68
VGGNet	76.0	68.4	79.2	65	61	65
ResNet	82.1	79.2	83.1	72	65	79
DenseNet	76.0	79.7	83.0	69	71	73
MDNet	83.4	80.1	85.2	76	73	81

TABLE I: Overall classification accuracies and average class precision (ACP) for different models on MLC dataset.

open source implementation of CF, we only report results on the MLC dataset as the authors [11] showed in their work. We also evaluate the classification performance of other state-of-the-art algorithms: DenseNet [8], ResNet [7] and VGGNet [6], by using the corresponding authors' open source implementation. We evaluated different architectures of these state-of-the-art algorithms. However, we choose the following architectures to report as they performed best against their variations: DenseNet with 121 layers, ResNet with 50 layers, and VGGNet with 16 layers. We used pre-trained weights to train ResNet and VGGNet models, since the dataset is quite small compared to the number of hyper-parameters of these networks.

B. Classification Performance

The overall classification accuracy and precision on the MLC dataset is shown in Table I. Our method significantly outperforms the baseline networks on MLC dataset. For both ResNet and VGGNet architectures, we initialize the networks with ImageNet pre-trained weights. We trained both ResNet and VGGNet models in two setups. In the first setup, we trained all the parameters of the networks using the MLC dataset [2]. In the second setup, we only trained the fully connected layers of the networks while all previous layers remained fixed on the pre-trained weights. For VGGNet and ResNet, fine-tuning all the hyper-parameters provided the best result. Therefore, we only report the performance of VGGNet and ResNet in the second setup. For DenseNet, we choose the first setup, that is, to fit all the parameters of the network on the dataset. Therefore, we initialize the DenseNet model with ImageNet pre-trained weights and then train the entire model.

CF is built utilizing the extracted feature representation from VGGNet's last layer before the first fully connected layer. So, the performance gain by CF is due to the use of several patch sizes and combination of hand-crafted features. Moreover, the accuracy gain of the CF is lower than the state-of-the-art models in some cases. Our proposed model outperforms these models and achieves 83.4% accuracy for Experiment 1, 80.1% accuracy for Experiment 2, and 85.2% accuracy for Experiment 3. MDNet achieves competitive accuracy gain over the baseline methods because it uses different size patches along with cost sensitive training.

Table I reports also the average class precision for all the models. We observe that our proposed method achieves significantly higher average precision than other baseline models. The result indicates that the proposed network generalizes well, and predicts the less frequent classes more accurately.

V. Conclusion

In this paper, we presented a novel approach, termed MDNet, for coral classification which outperforms the state-of-the-art algorithms in a standard benchmark dataset for coral species classification. We devised a method to effectively tackle point annotated datasets and make them compatible for use in a CNN. In addition, we improve classification performance in the presence of the class imbalance problem in coral datasets by utilizing implicit deep supervision and employing cost sensitive training of the proposed network.

Future work includes designing an online coral annotation method capable of detecting corals on embedded computers. This online coral annotation method will enable autonomous monitoring of coral reef health using different underwater robots. Currently, in the proposed work, we specify the input size for the network. Such specification may not produce the best results when tested on different types of data with point annotation. Therefore, we are currently exploring the possible application of reinforcement learning to automate the selection of patch size for a dataset. The deployment of the proposed trained network on a Neural Computation Stick will enable the utilization of the proposed technique by scientists in the field at a minimal cost.

References

[1] F. Rohwer, M. Youle, and D. Vosten, *Coral reefs in the microbial seas*. Plaid Press United States, 2010, vol. 1.

[2] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1170–1177.

[3] O. Beijbom, P. J. Edmunds, C. Roelfsema, J. Smith, D. I. Kline, B. P. Neal, M. J. Dunlap, V. Moriarty, T.-Y. Fan, C.-J. Tan *et al.*, "Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation," *PloS one*, vol. 10, no. 7, p. e0130312, 2015.

[4] T. Manderson, D. Meger, J. Li, D. C. Poza, N. Dudek, and G. Dudek, "Towards autonomous robotic coral reef health assessment," in *Proc. of Field and Service Robotics (FSR)*, Toronto, Canada, 24-26 June 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[10] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[11] A. Mahmood, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, G. Kendrick, and R. Fisher, "Coral classification with hybrid feature representations," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2016, pp. 519–523.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *Int. Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[13] A. Mehta, E. Ribeiro, J. Gilner, and R. van Woesik, "Coral reef texture classification using support vector machines." in *Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, 2007, pp. 302–310.

[14] M. S. A. Marcos, L. David, E. Peñaflor, V. Ticzon, and M. Soriano, "Automated benthic counting of living and non-living components in ngedarrak reef, palau via subsurface underwater video," *Environmental monitoring and assessment*, vol. 145, no. 1-3, pp. 177–184, 2008.

[15] M. D. Stokes and G. B. Deane, "Automated processing of coral reef benthic images," *Limnology and Oceanography: Methods*, vol. 7, no. 2, pp. 157–168, 2009.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. on Machine Learning (ICML)*, 2015, pp. 448–456.

[17] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 315–323.

[18] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri, "Cost sensitive learning of deep feature representations from imbalanced data," *arXiv preprint arXiv:1508.03422*, 2015.

[19] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.