Eng Int Syst (2021) 4: 253–261


**Engineering**
*Intelligent Systems*

# Automated CNN Based Coral Reef Classification Using Image Augmentation and Deep Learning

**Sumit Sharan[1], Harsh[1], Stuart Kininmonth[2] and Utkal Mehta[1]***

[1] *School of Engineering and Physics, The University of the South Pacific, Laucala campus, Suva, Fiji*
[2] *The University of Queensland, Heron Island Research Station, Qld Australia*

A critical issue faced by the marine scientist is to classify underwater images describing coral benthic cover. Typically, scientists take underwater imagery using high-resolution cameras and further analysis on these corals and marine species is done on land (preferably a laboratory) and by visual inspection. However, the analysis is time consuming, since the first step, which is the classification of corals, is an intensive activity by taxonomic experts. This traditional manual classification method is difficult to automate or quicken which is problematic given the high volume of images. In this work, the fundamental analysis is discussed by using available techniques such as deep learning (DL) and Convolutional Neural Network (CNN). It is required to find an easier, efficient and faster way to automate the classification of corals. This task is complicated, since most of the common coral species look similar to one another. For reasons of structural diversity, it is easier to differentiate other forms of marine life such as fish and stingrays. This paper is based on the difficult but important Scleractinian (Stony) corals only. A technique recommended is investigated further at structural level such as branching corals. Verification result proves that the training and testing data are almost similar, thus the proposed technique is capable to learn and predict correctly.

Keywords: Coral classification, CNN, Automation, Deep learning, RGB approach.

## 1. INTRODUCTION

Artificial Intelligence (AI) is being regularly modified with different approaches that aim to maximize the benefit. This literature review looks at the past approaches that have been utilized and gives a clear overview on the most suitable approach for the successful classification of corals. AI can process large amount of data at the same time and therefore it is commonly referred as the parallel processing [1]. When complex algorithms are introduced, the path to achieving the set goals gets easier and the process becomes faster. The infrastructure of computations is entirely based on the perceptions of the hierarchy. The reason for this process being called DL is due to the multiple computing layers that are involved with the processing of simple concepts.

*Corresponding address: Email: utkal.mehta@usp.ac.fj

There is a wide range of AI applications that are used nowadays. The type of application determines the cost as well as the effectiveness of the project. In this study, we focus mainly around the DL approach for coral classification. In this area, AI is being used to recognize coral images and classify them accordingly. This can later be categorized as species, family type or even the density of the coral in a particular area [2]. The AI is widely being used presently on mobile based applications, which clearly means that the growing technology is reaching out straight to the palms of the end users. On this topic, the Convolutional Neural Network (CNN) is one of the efficient approaches to developing AI systems. This method is mostly used to solve very complex problems. Nevertheless, this is due to CNN overcoming the limitations that are a problem when traditional machine learning methods are used. The architecture used in CNN is known as feed-forward. Perceptron's are computer models that represent the ability of the AI to specifically recognize data, is arranged in a number of layers, where the input is given to the first layer and the output is taken from the last layer. The layers that are in between these two outer layers are usually "hidden" since these layers do not have any connection with the outside feeding data [3].

In the network, each perceptron in a single layer has a connection with every other perceptron, so when the data is fed to the system, it goes in all the possible paths on its way, right to the output layer/perceptron. Therefore, it is called the feed forward architecture. It can learn from the extracting features and from a single picture in a very short period. Moreover, it can identify different objects very efficiently. There are other reasons why CNN is preferred over other alternate means, discussed in [4]. It is reported that it works on weight sharing concept, which results in less number of parameters to be trained. When the parameters are less, the training of the CNN can be done smoothly and it does not have to go through overfitting. Moreover, the stage at which the network classifies is incorporated with the stage where the features are extracted, and both of these stages use the learning process.

This paper firstly reviews the various techniques based on the extraction of features of a single type of coral. The extraction features that can be extracted from an image are on the classification type. As mentioned earlier, the classification type could be on the family, species or the color of the corals. Coral surface texture, types of pores and the length of the branches are other supporting features which can enable a machine to accurately classify the corals. It is desired to automate the classification of corals efficiently without adding more computational complexity. Though this is intricate, because the common coral species look similar to one another. This paper is focused on the challenging tasks for group of corals namely Scleractinian (Stony). A technique is examined further at structural level such as branching corals. Real verification result shows that the training and testing data are almost similar, thus the proposed technique is capable to predict correctly with acceptable identification accuracy.

## 2. LITERATURE REVIEW

This is a key point in any method on feature extraction when it is incorporated together with the CNN. What the network does in this case is it uses a number of dots, depending on the designer, and checks for the same patterns on the data. A system can be trained with this information. This makes the process of classification even easier and faster. The higher the number of dots, the more time it takes for classification. However, with the increased number of dots, the accuracy also increases. In 2017, the researchers presented the feature extraction process with DL [5]. The whole process has been classified as Contemporary Convolutional Network in 1980 by a researcher Fukushima and known as Neocognitron [6]. It was perhaps the first ANN that was collaborated with DL and this is a well-known approach in terms of DL algorithm.

However, due to the increased number of layers that arose from the feed-forward architecture, it was later hypothesized that the system could not process the multiple numbers of parameters that were set for an image classification. This was also due to Namatevs using the manual training methods of the algorithm. Thus, it could be seen that the obtained results did not entirely satisfy the needs/objectives that were initially set for the project. In 1986, a back-propagation method was invented which involved one or two hidden layers to train a network. Based on the training the network, it can able to accurately classify the different images. The functionality of such network type was used to see how it can prove to be viable for the application of classification and identification. The technique involved a hierarchy comprising of shift invariant feature detectors that work only for image classification. This was then applied the same year to Neocognitron, which gave some unrealistic results. This meant that the approach was not that viable for such an application. This application was in reference to weight sharing, and to the layers of convolutional neurals, with some adaptive connections.

In 1991, Fallside and Robinson proposed an RNN (Recurrent Neural Network) for the purpose of speech recognition. For speech recognition, a multilayer perceptron should be used. This makes the recognition of speech clearer for analysis when passed through multiple layers. This approach that has started in 1980 came to an end in 1995, when the projects that used this approach failed to work, or operate as expected. Later in 2006, Hinto [7] showcased a type of neural network that was believed to overthrow Gaussian or the RBF (Radial Basis Function) kernel. This was on the MNIST benchmark. These GPUs were integrated with CNN compatibility and were seen to be approximately 4 times faster than the common Computing Processing Units (CPUs) or now known as Central Processing Units. Some of the critical components that are now associated with deep CNN include GPUs, FPGA (Field Programmable Gate Arrays), DSP (Digital Signal Processing) and other silicon-based architectures which enable the user to get computer-based operations done.

Let the primary convolutional operation is written as below, which was adapted by almost all researchers for convolution.

$$g(x) = f(x) \otimes h(x) = \int_{-\infty}^{\infty} f(s)h(x-s)ds \tag{1}$$

It is known that $f(x)$ and $h(x)$ are two functions, while $s$ is denoted as a dummy variable for integration, which can take the value of either a 0 or a 1. For the two-dimensional convolutional, the two given functions are given as:

$$g(x, y) = f(x, y) \otimes h(x, y) = \iint_{\infty} f(s, t)h(x-s, y-t)\text{ds dt} \tag{2}$$

However, convolution and multiplication are two distinctive operations that are performed for different application purposes. In a one-dimensional application, the system is in a time domain, which is similar to $x(t)$ and for some cases, in the frequency domain $w(a)$. With the known basics, the convolution operation is further given by,

$$s(t) = (x \otimes w)(t) \tag{3}$$

where, $x$ is referred to as the first argument input, $w$ as the second argument (kernel) and $s(t)$ as the kernel map or the output. As previously mentioned, the algorithm automatically discretizes the whole system; therefore, the value of $t$ will only be able to take whole values in order to be processed. The discrete system is then defined by:

$$s(t) = (x \otimes w)(t) = \sum_{-\infty}^{\infty} x(a)w(t-a) \quad (4)$$

The input that is usually given to the application is data, which is made up of a multidimensional array while the kernel is a parameter that is also a multidimensional array. These parameters are then picked up by the algorithm which is further processed in the network. The multidimensional arrays are also known as tensors. The two-dimensional kernel and space convolution is given as,

$$s(i, j) = (I \otimes K)(i, j) = \sum_{m} \sum_{n}^{I(i-m, j-n)K(m,n)} \quad (5)$$

If it is assumed that there are fewer variations of the range that the values of $m$ hold, when compared to the values of '$n$', assuming that the convolutional operation is commutative, (5) can be re-written as follows:

$$s(i, j) = (K \otimes I)(i, j) = \sum_{m} \sum_{n}^{I(i-m, j-n)K(m,n)} \quad (6)$$

For instance, if $m$ is said to increase, the index that goes to the input also increases, however the index of the kernel would decrease. This means that the kernel that is relative to the input has been flipped. If the kernel is seen to not have flipped, the following function (known as cross-correlation) can be used.

$$s(i, j) = (K \otimes I)(i, j) = \sum_{m} \sum_{n}^{I(i+m, j+n)K(m,n)} \quad (7)$$

In regards to machine learning, the algorithm will be able to learn the different values of the kernel at the places where it is appropriate. For AI systems, machine learning has to be used together with other functions. If used on its own, the algorithm may be seen to malfunction and not provide expected results. Technically, a new means of data processing known as DL has been introduced. This allows AI systems to be developed for many routine tasks. This tackles the limitations in big data processing in the computer systems industry. The AI is used to handle problems from as simple as performing basic calculations to complicated problems like image processing. However, there are also many other problems that AI's are used to solve. This includes recognizing drawings in an image, recognizing vocal patterns and many more. They are known for their flexibility and performance. Nevertheless, the DL algorithms have broken accuracy records in classification of images [1] as well as speech recognition [8]. The DL is known to be developing, expanding and changing radically every day, and affecting the way that we use the power of computers in everyday lives. This power can be used to detect features that are very specific in data and furthermore use that data for other purposes such as classification, to create predictive models or clustering [9].

The application of these tools in regards to AI have taken over most parts of the world for automating tasks such as maneuvering a driverless car on the road [10] or smart path guidance [11]. In addition, these tools are also being applied in the medical field to detect breast cancer [12], finance [13], bioinformatics [14] and some more applications presented in the literature. Even some popular video games that are being played nowadays also use the DL [15].

From literature studies, it is obvious that the AI tools with DL can potentially help for solving more complicated problems in areas like ecology and specifically coral classification. Application of the DL in ecology can prove to be viable as DL has the ability of processing complicated data in a short amount of time. These data are suitably nonlinear and enables the AI to overcome the complexity of missing information. It is presented well in [16]. Moreover, the machine learning is well adopted already in the area of ecology to manage operations such as ecological modeling. The application such as speech recognition and teaching language using emotion mining and deep learning algorithms is very successful [17]. Similarly, it is interesting to study the behavior of certain animals [18]. It can be seen that a self-learning approach is also useful in motion and posture control application [27].

In summary, a self-learning feature of DL makes it better than other approaches, without external input being provided. There exist two distinct means of training machines. One of these methods is without the need of supervision. This is where computers are able to detect patterns automatically, as well as the similarities between unlabeled data, and in this case, images. When this method is used, a specific output cannot be expected, since the output is dependent on the type of training the machines goes through, and the features that it learns could differ each time. This leads to the system being used as an exploratory tool that can be used to detect certain features in a set of data, cluster other groups that are similar, or possibly reducing the number of dimensions. Learning with supervision is done in cases where prediction tasks, detection, and identification is involved. The computer is fed with a dataset that is labeled with the objects that are to be identified. The computer then takes the dataset and processes it while learning the feature by itself. It then stores these features on a file and recalls if every time it needs to access those features. After training is completed, it is able to recognize the same images, of which the dataset was provided. The computer is then able to identify the objects from the dataset.

Another method to train machines is widely known as CNN. This method does not only work with provided labels within datasets, but the user has to specifically program the system on the features to extract and what other features to look for. For instance, if a giraffe has to be detected in a picture, the characteristics of a giraffe will have to be programmed in order for the algorithm to be able to detect a giraffe in a picture. This has a downside in areas where the programmer has the least amount of knowledge associated with the problem statement, or the area the DL is being used in. In overview, DL methods skip these procedures. The algorithms of DL automatically detect features on an image and extract those features. This leads to a programmer telling the algorithm if a giraffe is actually present in a certain image, and when ample examples are given to the system as training, it is able to uniquely identify if an image has a giraffe present. The available data is then decomposed in to multi layers, which have to be created first. The different layers allow the algorithm to learn different features from the layers that are made available to it. This ability of DL to automatically detect features in very high dimension data, complex and a high predictive accuracy makes it an everyday expanding technology in regards to DL methods.

The DL is believed to be a key tool for many ecological analyses applications [19]. However, the mathematical complexity and the massive programming skills that is required in order to develop such a tool is intimidating and may prevent ecologists from using the tool. To add on, the approach of automatic classification of corals may free up experts from doing the tedious work of manual classification. It may also enable faster processing and repetitive annotation of ambiguous images. As mentioned earlier, deep networks involve multi-layer processing which are composed of both linear and as well as non-linear operations. For a particular problem to be solved, the certain parameters of that particular problem are learned. Previously, some representations in image forms have been extracted from a dataset, known as ImageNet [20] have proven to give reasonable results in terms of identification, classification and recognition operations [10, 14, 21, 22]. If the CNN is made limited to the resolution of images that it is provided with, it complicates the whole process. Therefore, it is known that "Spatial Pyramid Pooling (SPP) [23] and "Multi-Scale Orderless Pooling (MOP) [18] are two methods of pre-processing an image before it is passed on to the CNN part. This enables CNN to disregard the size of the image that is being given as input to it, thus, narrowing down the process. For better results, the algorithm has to be trained on a large dataset such as ImageNet and following this, it has to be fine-tuned on specific types of features.

Coral reefs are an important part of the marine ecosystem. There are continuous reports of decline in the heath and quality of corals. With the help of AUVs (Autonomous Underwater Vehicles) [24] and diver sleds [25], which are towed provide a good amount of underwater data that is utilized for analysis. The ground level spectroscopy used for coral reefs classification using hyperspectral camera [26], however such approach is expensive and prolonged. The automatic annotation of underwater imagery is a challenging task because as known, corals have ambiguous shape, color and texture. The turbidity of water and the illumination that is present underwater affect the quality of the images that are taken to be processed [24].

In the next section, the proposed approach is discussed with objective to improve the classification of coral from simple images. The method is simple yet automatic estimate the coral type. The primary method is discussed with real time data using the convolutional neural network.

## 3. PROPOSED APPROACH

Before we discuss the proposed approach in details, let us understand the briefly the steps from Fig. 1 It states the use of image augmentation techniques and RGB processed images for training. The training set of images was collected by marine experts and were manually annotated and categorized into the different coral families. These images were then used to train the neural network. There were two approaches in this paper. The techniques were analyzed and the best performing was applied to train the neural network. Now, in following subsections, each process step is elaborated in detail.

### 3.1 Grayscale Approach

In the presented work, we have applied image augmentation (such as rotation, zooming, segmentation and resizing) in the processing. This allowed generation of a larger dataset. Once the image augmentation is performed, the images are then converted to grayscale. The dimension of the grayscale consists of height, length and only one layer as opposed to RGB whereby there are three layers. The images are resized to $150\times150$ pixels while performing the image augmentation. It is important to normalize the training dataset for the neural network. To note that the increase in the image dimension increases the computational power. Fig. 2 shows the example of each image segmentation, conversion to grayscale and RGB as discussed in next subsection.

### 3.2 RGB Approach

The RGB approach follows the same image augmentations as stated in the grayscale approach above. Since there are distinct colors and patterns of each coral family, the RGB approach is appropriate whereby the colors along with the coral patterns can be used for feature extraction and classification. The images are resized to $200\times200$ pixels. Then, the training set of images is converted to RGB. The dimensions of the resized images consisted of height, length and the three layers, which are, red, green and blue. Since the width consists of three layers, the processing of RGB images requires more computational power when compared to the grayscale approach. The training set is prepared and then used to train the neural network.

### 3.3 Final Pre-Processing

After creating dataset, the team trained the neural network with 10 types of corals with each coral type having 200–220 images for training and 20 images each for prediction. Since some of the images had to be cropped in order to obtain just a single coral in an image, this reduced the overall image pixel size therefore after slicing, the image size was reduced to $200\times200\times3$ pixels in order to have a definite constant image size which is fed to the neural network.

## 4. REALTIME CLASSFICATION AND RESULTS

After the preprocessing of images, the set of images can be passed through the neural network. A total of 10 epochs was used in the experiment required transit the dataset through the neural network back and forth. This is the training of the network with the dataset. In our study, each category of image,

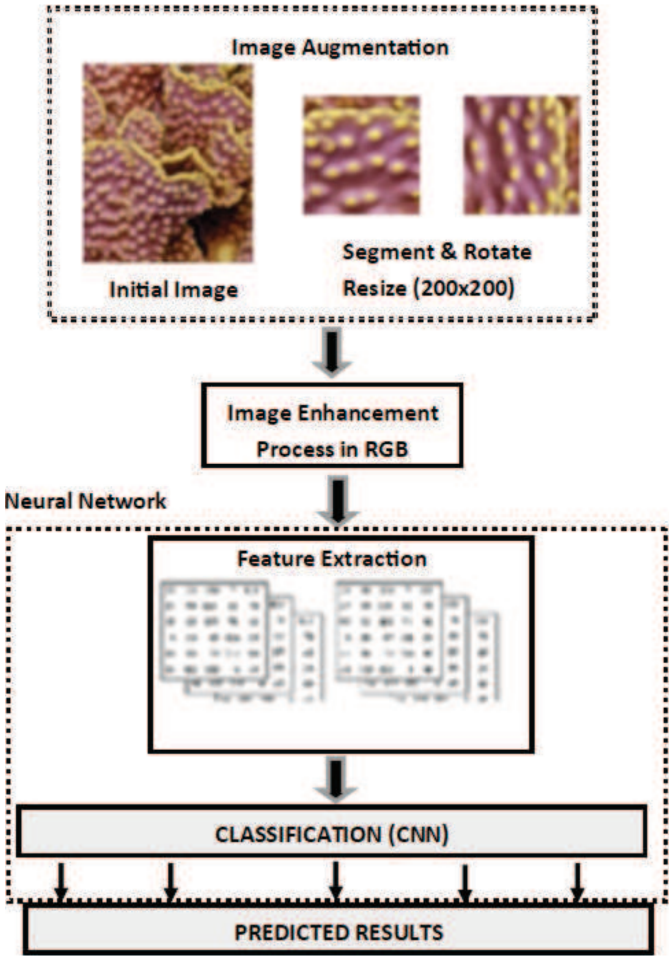



**Figure 1** Flowchart for the Presented Technique.

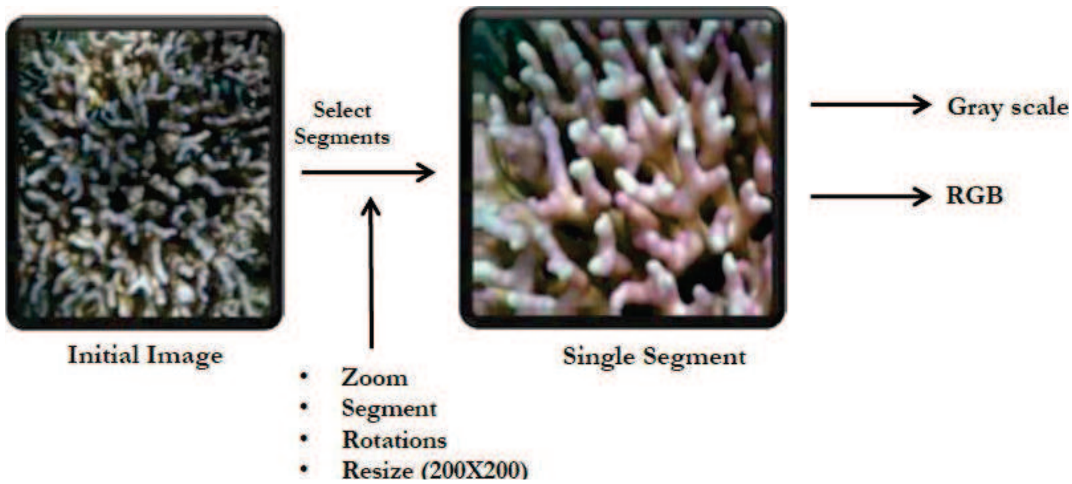
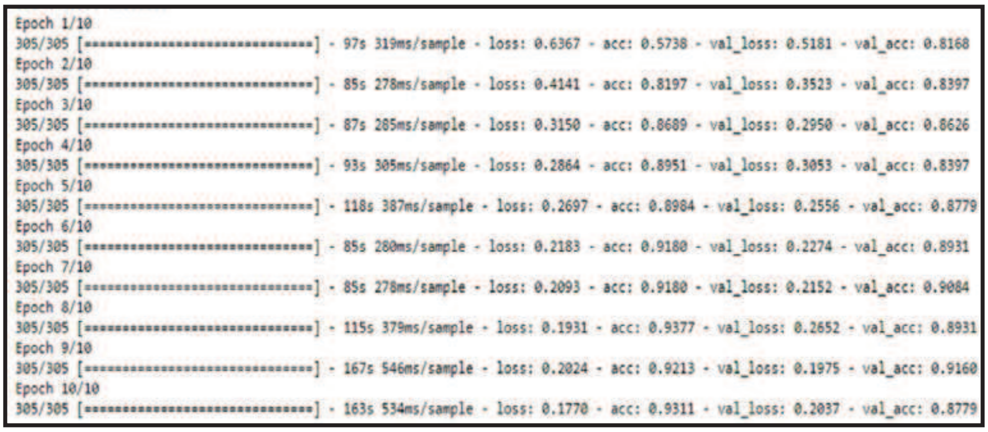



**Figure 2** Image Pre-processing layout.

**Table 1** Comparison of the Two Methods.

| Method | # of Epochs | Accuracy |
|---|---|---|
| RGB | 10 | 94.5% |
| Grayscale | 10 | 93.0% |

that is, hard coral and soft coral had 233 images, adding to a total of 466 images. The Epoch procedure took 305 images for training and 161 images for validation. After the 10 Epochs, approximately 93% accuracy was acquired. The epoch result was obtained and is shown in the snippet in Fig. 3. A total of 10 epochs were done and the parameters included were the



```
Epoch 1/10
305/305 [==============================] - 97s 319ms/sample - loss: 0.6367 - acc: 0.5738 - val_loss: 0.5181 - val_acc: 0.8168
Epoch 2/10
305/305 [==============================] - 85s 278ms/sample - loss: 0.4141 - acc: 0.8197 - val_loss: 0.3523 - val_acc: 0.8397
Epoch 3/10
305/305 [==============================] - 87s 285ms/sample - loss: 0.3150 - acc: 0.8689 - val_loss: 0.2950 - val_acc: 0.8626
Epoch 4/10
305/305 [==============================] - 93s 305ms/sample - loss: 0.2864 - acc: 0.8951 - val_loss: 0.3053 - val_acc: 0.8397
Epoch 5/10
305/305 [==============================] - 118s 387ms/sample - loss: 0.2697 - acc: 0.8984 - val_loss: 0.2556 - val_acc: 0.8779
Epoch 6/10
305/305 [==============================] - 85s 280ms/sample - loss: 0.2183 - acc: 0.9180 - val_loss: 0.2274 - val_acc: 0.8931
Epoch 7/10
305/305 [==============================] - 85s 278ms/sample - loss: 0.2093 - acc: 0.9180 - val_loss: 0.2152 - val_acc: 0.9084
Epoch 8/10
305/305 [==============================] - 115s 379ms/sample - loss: 0.1931 - acc: 0.9377 - val_loss: 0.2652 - val_acc: 0.8931
Epoch 9/10
305/305 [==============================] - 167s 546ms/sample - loss: 0.2024 - acc: 0.9213 - val_loss: 0.1975 - val_acc: 0.9160
Epoch 10/10
305/305 [==============================] - 163s 534ms/sample - loss: 0.1770 - acc: 0.9311 - val_loss: 0.2037 - val_acc: 0.8779
```

**Figure 3** Loss and accuracy results from trial.

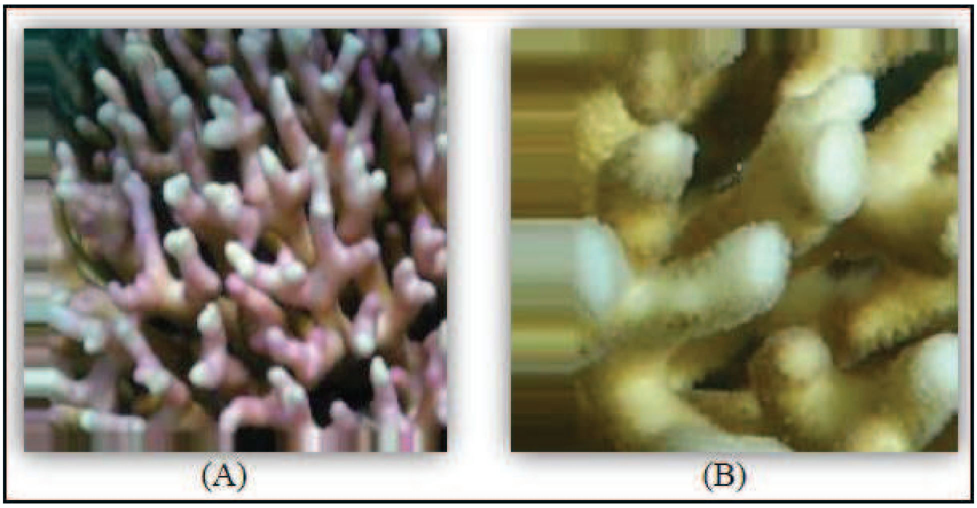



**Figure 4** Types of Images Used for Training.

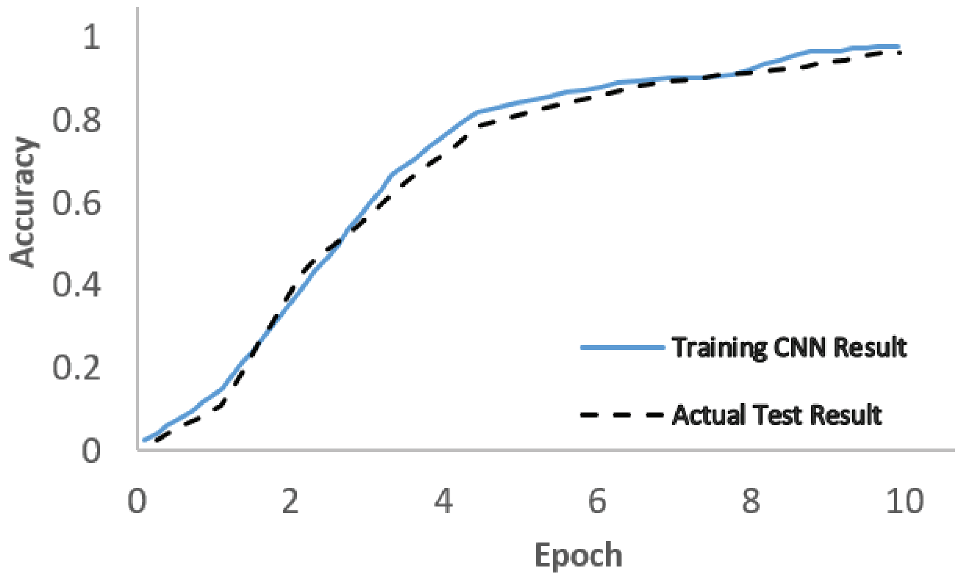



**Figure 5** Graph of Epoch Accuracy.

training loss, accuracy and the validation loss and accuracy. An ascending order can be noticed in the accuracy column and a descending order in the loss column, this means that the training is accurate.

In this way, the training dataset was created by segmenting multiple images from a larger image and then applying rotations and zooming by a factor of 0.2. The training set consisted of a combination of the 10 different types or

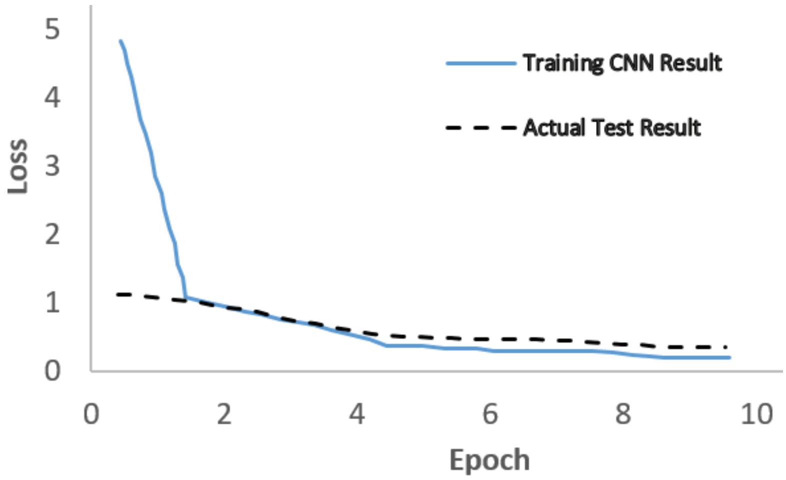



**Figure 6** Graph of Epoch Loss.

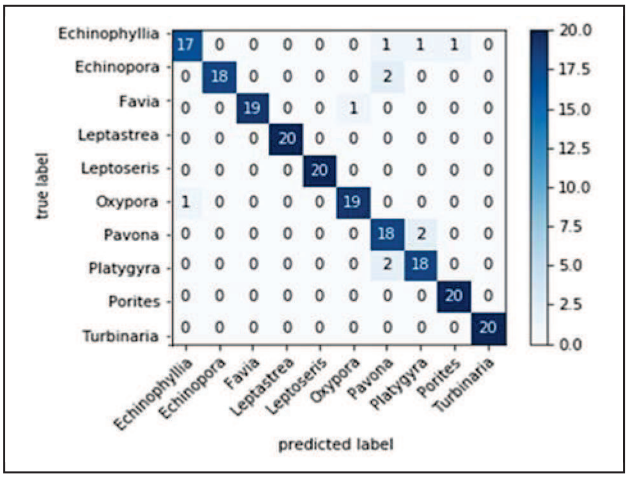



**Figure 7** Confusion Matrix for RGB Approach.

coral images. There were approximately 220 images for each coral type. This was done to balance the images for each coral. In this way, approximately 2200 images were used for training and verification purpose. The testing set of images consisted of 50 images for each coral type, thus a total of 500 images used for testing. Fig. 4 (A) shows the actual image and (B) shows the zoomed in image. Note that image augmentations are applied to the images. Each image was zoomed in by a factor of 20% in order to make the patterns visible and easier for the neural network to extract. Then the training set of images was converted to RGB. The RGB approach achieved better results when compared to the grayscale pre-processing approach.

## 4.1 Validation

It was important to understand the difference between the grayscale and RGB pre-processing methods. The grayscale approach mainly became a 0 and 1 type of scenario whereby only the shape of the coral was seen by the neural network which can be the black portion and the other portion being white. The RGB method analyses and convolves each layer R, G and B along with the distinct patterns of each coral type. Average pooling was used in order to obtain the common information. A two-dimensional convolution layer was used along with Rectified Liner Unit (ReLu) yielding the best results. Since multiclass classification was required, a softmax layer was used which was able to predict each class with a given probability on a scale of 0 to 1. The neural network composed of 12 layers along with the use of dropout layers which prevents over-fitting while training. The neural network was created and optimized by adding the various layers along with having a proper training and validation split.

In Fig. 5, the epoch accuracy graph is shown. The training and testing graphs are almost similar which indicates that the model was able to learn and predict quite accurately. Fig. 6 shows the graph of epoch training and testing. It can be seen that the loss decreases, which is the ideal outcome of a model. Observing Fig. 5 and 6, the model was not over-fitting or under-fitting and therefore, it was accurately trained and the parameters were learned efficiently.

The best performance is achieved by the use of the RGB approach. In order to assess the performance of the trained model, new set of images is tested for prediction, which are not included in the training process. Fig. 7 shows the confusion matrix, which consists of the 10 coral, types and can be observed that there are 4 classes for which the model predicted all images accurately. The other classes prediction accuracy ranging from 85% to 95%. The overall model accuracy was found to be 94.5%. It is proved that the majority of the predictions has been resulted correctly. However, when the structure of some corals is complex and very similar corals, the prediction is not correct.

## 5. CONCLUSIONS

The simple yet effective automatic classification technique for corals is presented in this paper. The model has been trained by using RGB and grayscale approaches. Considering the results obtained from the RGB approach, the accuracy of 94.5% was achieved for classifying ten different types of corals. The RGB approach for pre-processing was better, when compared to the grayscale approach. Due to the complexity in the structure of some corals and similarities between the corals, the prediction for some of the types of corals were incorrect, however majority of the predictions had been resulted correctly for new set of data. In future work, the number of images per coral should be increased, this way one can see better prediction process. The increasing number of images per coral means an increased variety of observations being made and parameters being learnt by the network. Nevertheless, there is no definite answer as to how big a dataset should be and depends on the type of application, quality of images and the size of images for training. The more the data, the better it is for statistical analysis of the model. Simply, a bigger dataset is advantageous. The model can be trained with more types of corals in order to obtain automated tool, which can be used by marine scientists or individuals for coral annotation.

## REFERENCES


1. Malhotra, Y. (2018). AI, Machine Learning & Deep Learning Risk Management & Controls: Beyond Deep Learning and Generative Adversarial Networks: Model Risk Management in AI, Machine Learning & Deep Learning, *SSRN Electronic Journal, 6*(3): 12–14.
2. Qi, Z., Wang, B., Tian, Y. and Zhang, P. (2016). When Ensemble Learning Meets Deep Learning: A New Deep Support Vector Machine for Classification, *Knowledge-Based Systems, 107*(3): 54–60.
3. Razavian, S., Azizpour, H., Sullivan, J. and Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition, in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 512–519.
4. Oquab, M., Bottou, L., Laptev, I. and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks, in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1717–1724.
5. Namatēvs, I. (2017). Deep Convolutional Neural Networks: Structure, Feature Extraction and Training, *Information Technology and Management Science, 20*(1).
6. Fukushima, K. (2015). Neocognitron: Deep Convolutional Neural Network, *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, 27*(4), 115–125.
7. Hinton, G. (2006). To Recognize Shapes, First Learn to Generate Images, Technical Report UTML TR 2006-003, University of Toronto, 2006.
8. G. Hinton *et al.*, (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups, in *IEEE Signal Processing Magazine, 29*(6): 82–97.
9. Olden, J., Lawler, J. and Poff, N. L. (2008). Machine learning methods without tears: A primer for ecologists. *Q. Rev. Biol.* 83, 171–193.
10. Memon, M., Maheshwari, M. Shin, D., Roy, A. and Saxena, N. (2019). Deep-DRX: A framework for deep learning-based discontinuous reception in 5G wireless networks, *Transactions on Emerging Telecommunications Technologies, 2*(4), e3579.
11. Mehta, U., Alim, M. and Kumar, S. (2017). Smart path guidance mobile aid for visually disabled persons, *Procedia Computer Science, 105*, 52–56.
12. Golden, J. A. (2017). Deep learning algorithms for detection of lymph node metastases from breast cancer: Helping artificial intelligence be seen, *JAMA* 318, 2184–2186.
13. Heaton, J. B., Polson, N. G., and Witte, J. H. (2016). Deep learning for finance: deep portfolios, *Appl. Stoch. Models Bus. Ind. 33*, 3–12.
14. Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Brief. Bioinform. 18*, 851–869.
15. Lample, G., and Chaplot, D. (2017). Playing FPS Games with Deep Reinforcement Learning, in *ThirtyFirst AAAI Conference on Artificial Intelligence*.
16. Thessen, A., (2016). Adoption of machine learning techniques in ecology and earth science, *One Ecosyst, 1*, e8621.
17. Zhang, X., Li, H., Wang, N., and Shi, R., (2019). Optional English Speech Teaching Method Based on Recognition Emotion Mining and Deep Learning Algorithms, *Engineering Intelligent Systems, 27*(3): 141–150.
18. Gong, Y., Wang, L., Guo, R. and Lazebnik, S. (2014). Multi-scale orderless pooling of deep convolutional activation features, in *Computer Vision– ECCV 2014*, 392–407, Springer, 2014.
19. Rovero, F., Zimmermann, F., Berzi, D., and Meek, P. (2013). Which camera trap type and how many do I need?' A review of camera features and study designs for a range of wildlife research applications. *Hystrix Ital. J. Mammal. 24*, 148–156.
20. Deng, J., Dong, W. and Socher, R. (2009). Imagenet: A large-scale hierarchical image database, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255, IEEE, 2009.
21. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition, *International Conference on Machine Learning (ICML)*, 647–655.
22. Beijbom, O., Edmunds, P. J., Kline, D., Mitchell, B. G., and Kriegman, D. (2012). Automated annotation of coral reef survey images, in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012*, 1170–1177.
23. He, K., Zhang, X., Ren, S. and Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition, in *European Conference on Computer Vision (ECCV)*, pp. 346–361, Springer, 2014.
24. Patterson, M. R. and Relles, N. J. (2008). Autonomous underwater vehicles resurvey bonaire: a new tool for coral reef management, in *Proceedings of the 11th International Coral Reef Symposium*, 539–543.
25. Kenyon, J. C., Brainard, R. E., Hoeke, R. K., Parrish, F. A. and Wilkinson, C. B. (2006). Towed-diver surveys, a method for mesoscale spatial assessment of benthic reef habi- tat: a

case study at midway atoll in the Hawaiian archipelago, *Coastal Management, 34*(3): 339–349.
26. Caras, T. and Karnieli, A. (2013). Ground-level spectroscopy analyses and classification of coral reefs using a hyperspectral camera, *Coral Reefs, 32*(3): 825–834.
27. Sharma, B., Vanualailai, J. and Singh, S. (2017). Motion Planning and Posture Control of Multiple n-link Doubly Nonholonomic Manipulators, *Robotica, 35*(1): 1–25.