Comparative Analysis of Several Deep Learning CNN Architecture Models in Identifying Types and Health of Coral Reefs

Muhammad Aidan Daffa J - 1906300800
Dept. of Electrical Engineering, Computer Engineering Study Program
Faculty of Engineering, Universitas Indonesia
Kampus Baru UI, Depok 16424
Email: Muhammad.aidan@ui.ac.id

***Abstract*— Coral reefs are marine organisms that provide benefits to many other living creatures. The worsening pollution in the water and unpredictable climate changes threaten the health of coral reefs. Projections for 2050 indicate that 95% of coral reefs are likely to experience bleaching. This research proposes to apply deep learning to classify the types and health levels of coral reefs, with classifications divided based on the CoralWatch health chart, ranging from level 1 to 6. The health classification of coral reefs in this study is divided into 6 labels: lv.6, lv.5, lv.4, lv.3, lv.2, and lv.1. Meanwhile, for the classification of coral reef types, there are 3 classes: Boulder, Table, and Branching. The final outcome of this research is a model for classifying the types and health levels of coral reefs. The programming language used is Python, and the architectures used are ResNet, MobileNetV2, DenseNet, and VGG19. In this study, the best accuracy obtained for the classification of coral reef types is 100% with the DenseNet architecture, while for the classification of coral reef health, the accuracy obtained is 55% with the DenseNet architecture.**

***Keywords* — Machine Learning, Computer Vision, Convolution Neural Network, Image Classification, Coral Reef**

I. INTRODUCTION

Coral reefs are vital marine organisms in the life and ecosystem of the ocean. Beyond their role in the marine ecosystem, coral reefs also serve as a natural barrier, protecting coastlines from storms and erosion. However, due to damage caused by both natural and human factors, many coral reefs are dying and experiencing bleaching [1]. The health of coral reefs must be given attention due to the increasing water pollution and the ever-changing climate. Some scientists predict that if the current trends of human-induced pollution and climate change continue, 50% of coral reefs will be damaged by 2030. Therefore, this research focuses on the application of deep learning to assist in the classification of coral reef health based on the color chart provided by CoralWatch. The objective of this study is to develop a classification system for coral reef types and health using the CoralWatch chart, utilizing deep learning with the Convolutional Neural Network (CNN) method [2].

To ensure the availability of high-quality images for training the model and achieving better accuracy, a combination of internet-sourced datasets and offline collected datasets is used. This is done to ensure that a diverse set of high-quality images is used during the model training process.

In this paper, the researchers will explore and compare the performance of different CNN architectures, namely the MobileNetV2, DenseNet, ResNet-152, and VGG19. This could allow future studies to apply the result of experiments to further improve the findings, experiments, and advancements in the field of coral reef health and type identification.

II. RELATED WORKS

In the area of coral reef identification, several researchers have also studied the effectiveness of neural networks in learning health or type of coral reef. Utilizing CNN for determining coral health, Jon Borbon, Jeanette Javier, Jony Llamado, Elmer Dadios, and Robert Kerwin Billones develop models to identify a coral's state: (1) Healthy, (2) Dead and (3) Bleached. Proposed CNN model into 2 dataset from the RSMAS, dead corals from EILAT while bleached corals were taken from ReefBase. The results of the second dataset have an accuracy of 84.93% which is better than the first dataset that have an accuracy of 68.75%. The results showed that datasets which have larger sample size perform better than smaller datasets [1].

On the other hand, Ma. Shiela Angeli C. Marcos, Maricor N. Soriano and Caesar A. Saloma proposed to propose a model based on feedforward backpropagation neural network to classify close-up images of coral reef components into three benthic categories: living coral, dead coral and sand. This study have achieved a success rate of 86.5% (false positive = 6.7%) for test images. As inputs to the network this study derived from video stills of coral reef transects from the Great Barrier Reef. This paper also developed a rule-based decision tree classifier according to how marine scientists classify corals from texture and color, and obtained a lower recognition rate of 79.7% for the same set of images [3].

In conclusion above all the existing works of literature, the studies show that different neural network architectures have their own advantages and drawbacks, and the results will always be based on what feature-architecture combination suits the model and integrated database used. In this work, the researcher will try to explore which deep learning approach will give the best results when using a combined existing and natural database. Since almost all the existing studies focus on using existing databases, the researchers will investigate the effectiveness of different CNN architectures, namely the MobileNetV2, DenseNet, ResNet-152, and VGG19.

III. Methodology

The desired system is capable of recognizing the three types of coral reefs and the six levels of coral health. For this purpose, Neural Network technology is employed as the methodology. The Neural Network architecture utilized consists of Convolutional Layers to extract sequential features and learn information based on the provided feature sequence.

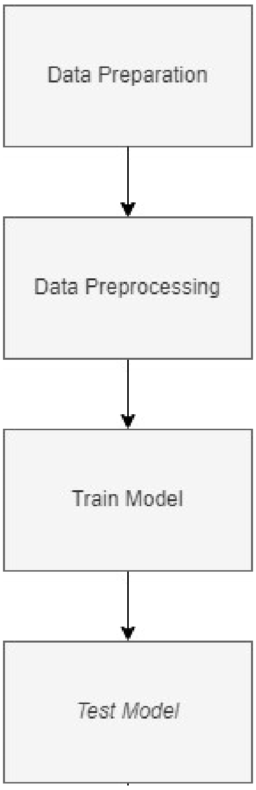

Fig. 1. Proposed Methodology

Fig. 1 shows the proposed methodology for the study. It comprises of four (4) phases: Dataset Preparation, Preprocessing, Train Model, and Test Model. This methodology was used to complete the experiments for the study.

A. Dataset Preparation

The data preparation process involves gathering data from both internet-sourced datasets and offline-collected datasets. The collected dataset will be divided into two parts: the training data and the testing data. The ratio between the training data and the testing data is 0.8 / 0.2.

Table 1. Collected Coral Reef Data

Coral Reef Type		**Coral Reef Health Level**	
Database	***Percentage***	***Database***	***Percentage***
StructuredRMAS	17.75%	StructuredRMAS	17.31%
Google Gambar	20.71%	Google Image	20.19%
Pulau Seribu	53.06%	Pulau Seribu	52.88%
Kaggle	8.48%	Kaggle	9.61%

The dataset used in this study was collected from various sources, including the StructuredRMAS database, Kaggle, manual Google image search, and direct capture in coral reef habitats (Thousand Islands). A total of 507 images were collected for the coral reef type dataset, and 520 images were collected for the health level dataset.

Table 2. Coral Reef Type Data Distribution per Label

Coral Reef Type	**Total**
Boulder	187
Branching	262
Table	58

Table 3. Distribution of Coral Reef Health Level Data per Label

Coral Reef Health Level	**Total**
Level 1	55
Level 2	81
Level 3	126
Level 4	120
Level 5	92
Level 6	46

Here are some examples of coral reef images obtained from the StructuredRMAS dataset, Google Images, Thousand Islands, and Kaggle:

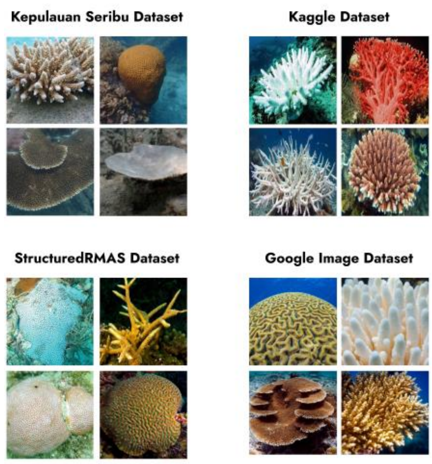

Fig. 2. Example Images of Coral Reef from Each Dataset

The dataset obtained from manual Google image search was collected by searching for specific keywords related to coral reef types, such as "Boulder Coral." As for the direct capture in coral reef habitats, the images were obtained from coral reef cultivators located in the Thousand Islands.

B. Data Preprocessing

The obtained dataset needs to be processed promptly before entering the training stage. Several steps need to be taken before the training stage, including performing one-hot encoding for each file's labeling, dividing the data into training and testing sets, and augmenting the training data.

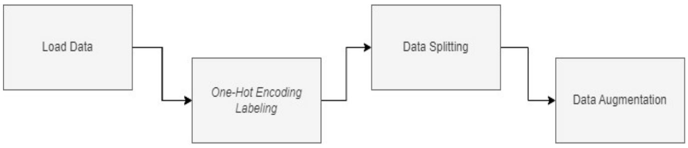

Fig. 3. Data Preprocessing

The process begins with loading the data, which will be directly stored in folders according to their data types. This storage method is implemented to facilitate data labeling. For the coral reef type classification problem, the files are divided into three folders: boulder, branching, and plate. Meanwhile, for the classification of coral reef health levels, the files are divided into six folders: level 1 to level 6.

Next, One Hot Encoding is applied to label the image data in each folder according to their folder names. After completing the data labeling process, the data is divided into training and testing sets. The ratio between the training data and testing data is 0.8:0.2. The final step before training is to create a function for augmenting the image data, which will be applied to the training data. This is done to increase the data variation based on the specified parameters.

C. Train and Test Model

In this research, the training of the models will be conducted to enable them to recognize the given dataset, thereby making predictions for the types and health levels of coral reefs. Four CNN architecture models will be used: MobileNetV2, ResNet, DenseNet, and VGG19. To obtain predictions from the employed deep learning models, a training process needs to be carried out using the dataset in the form of images, which were obtained after the preprocessing data.

- MobileNetV2

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenetv2_1.00_224 (Funct  (None, 7, 7, 1280)       2257984
 ional)

 average_pooling2d (AverageP  (None, 3, 3, 1280)       0
 ooling2D)

 flatten (Flatten)           (None, 11520)             0

 dense (Dense)               (None, 512)               5898752

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 50)                25650

 dropout_1 (Dropout)         (None, 50)                0

 dense_2 (Dense)             (None, 3)                 153

=================================================================
Total params: 8,182,539
Trainable params: 6,337,355
Non-trainable params: 1,845,184
```

The first model used is a transfer learning model with the MobileNetV2 architecture. The input consists of images of coral reefs to be classified into types or health levels using the MobileNetV2 model. In addition to the layers used in the base model of MobileNetV2, several additional layers are added to enhance the recognition and classification abilities of the initial MobileNetV2 model to predict the given images into predefined types or health levels.

- ResNet152

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet152 (Functional)      (None, 7, 7, 2048)        58370944

 average_pooling2d (AverageP  (None, 2, 2, 2048)       0
 ooling2D)

 flatten (Flatten)           (None, 8192)              0

 dense (Dense)               (None, 1024)              8389632

 dropout (Dropout)           (None, 1024)              0

 dense_1 (Dense)             (None, 1024)              1049600

 dense_2 (Dense)             (None, 512)               524800

 dropout_1 (Dropout)         (None, 512)               0

 dense_3 (Dense)             (None, 3)                 1539

=================================================================
Total params: 68,336,515
Trainable params: 11,020,291
Non-trainable params: 57,316,224
```

In the ResNet152 architecture, multiple convolution layers with various filters are used, as shown in Figure 4. The added layers include AveragePooling2D, Dense Layer with ReLU activation, dropout layers, two more Dense Layers, and another Dropout Layer. The final layer is a Dense Layer.

- DenseNet

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 densenet121 (Functional)    (None, 7, 7, 1024)        7037504

 average_pooling2d (AverageP  (None, 3, 3, 1024)       0
 ooling2D)

 flatten (Flatten)           (None, 9216)              0

 dense (Dense)               (None, 512)               4719104

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 3)                 1539

=================================================================
Total params: 11,758,147
Trainable params: 4,759,555
Non-trainable params: 6,998,592
```

In the DenseNet architecture, there are multiple convolution layers with various filters. The added layers include AveragePooling2D, Flatten Layer, and Dense Layer with ReLU activation. Dropout layers are also included, followed by another Dense Layer. The AveragePooling2D layer is used for downscaling the input images from the previous layers. This process aims to reduce the dimensions of the image and extract features. Then, a Dense layer is added to receive input from the previous layer's neurons. Before obtaining the prediction output, a dropout layer is included to randomly drop some neurons during the training process to prevent overfitting.

- VGG19

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg19 (Functional)          (None, 7, 7, 512)         20024384

 average_pooling2d (AverageP  (None, 3, 3, 512)        0
 ooling2D)

 flatten (Flatten)           (None, 4608)              0

 dense (Dense)               (None, 512)               2359808

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 50)                25650

 dropout_1 (Dropout)         (None, 50)                0

 dense_2 (Dense)             (None, 3)                 153

=================================================================
Total params: 22,409,995
Trainable params: 9,465,035
Non-trainable params: 12,944,960
```

In the VGG19 architecture, multiple convolution layers with various filters are also used, as shown in Figure 5. The added layers include AveragePooling2D, Flatten Layer, Dense Layer with ReLU activation, dropout layers, another Dense Layer, a Dropout Layer, and a final Dense Layer.

IV. EXPERIMENT AND RESULT

In this study, we conducted experiments on four models: MobileNetV2, ResNet, DenseNet, and VGG19. These experiments were carried out during the model training phase to compare the performance and training efficiency of these models. In this experiment, we compared the models based on the accuracy achieved, the time required for the training process, and other parameters.

The purpose of this testing phase is to determine which architecture among MobileNetV2, ResNet, DenseNet, and VGG19 performs the best in solving the coral reef classification problem. The testing is done by varying the model architectures while keeping other variables such as learning rate, number of epochs, and batch size constant.

Table 4. Experiment Parameters

Parameter	Value
Learning Rate	0.001
Epoch	20
Batch Size	8
Activation Function	Relu
Optimizer	Adam

The evaluation metrics used in this study are Accuracy, F1-Score, Recall, and Precision. Accuracy is used to measure the overall correctness of the predictions made by the models. F1-Score takes into account both precision and recall to provide a balanced evaluation of the model's performance. Recall measures the ability of the model to correctly identify positive instances, while Precision measures the proportion of correctly identified positive instances out of all instances predicted as positive.

A. MobileNetV2 Architecture Test

Here are the results obtained from the training experiment of the MobileNetV2 architecture for coral reef type and health level classification.

Table 5. Coral Reef Type Classification Report with MobileNetV2

	Precision	**Recall**	**F1 - Score**	**Accuracy**	**Training Time per Epoch**
Boulder	1.00	0.95	0.97		
Branching	1.00	1.00	1.00	98%	5 second
Table	0.86	1.00	0.92		

Table 6. Coral Reef Health Level Classification Report MobileNetV2

	Precision	**Recall**	**F1 - Score**	**Accuracy**	**Training Time per Epoch**
Level 1	0.83	0.29	0.43		
Level 2	0.00	0.00	0.00		
Level 3	0.22	1.00	0.37	81%	20 second
Level 4	0.00	0.00	0.00		
Level 5	0.00	0.00	0.00		
Level 6	0.00	0.00	0.00		

The MobileNetV2 architecture did not perform well in the classification of coral reef health levels. The results obtained show that it was unable to predict health level classes except for level 1 and 3. This could be due to the limited amount of data available, and when compared to the other classes, there is a significant difference in the number of samples. Another reason could be the lack of data variation and the architecture's inability to effectively handle this particular problem.

B. ResNet152 Architecture Test

Here are the results obtained from the training experiment of the ResNet152 architecture for coral reef type and health level classification.

Table 7. Coral Reef Type Classification Report with ResNet152

	Precision	**Recall**	**F1 - Score**	**Accuracy**	**Training Time per Epoch**
Boulder	0.71	0.12	0.21		
Branching	0.53	1.00	0.69	66%	12 second
Table	0.00	0.00	0.00		

Table 8. Coral Reef Health Level Classification Report ResNet152

	Precision	**Recall**	**F1 - Score**	**Accuracy**	**Training Time per Epoch**
Level 1	0.67	0.14	0.24		
Level 2	0.00	0.00	0.00		
Level 3	0.23	0.95	0.37	22%	33 second
Level 4	0.00	0.00	0.00		
Level 5	0.00	0.00	0.00		
Level 6	0.00	0.00	0.00		

The ResNet152 architecture performed poorly in the classification of coral reef types. The results obtained show that it was unable to predict the "table" class. This could be due to the imbalanced distribution of data compared to other classes. Additionally, the model was less capable of predicting the "branching" class. This may be attributed to the lack of data variation and the limitations of the architecture in handling this particular problem.

In terms of coral reef health level classification, the ResNet152 architecture also yielded unsatisfactory results, except for level 3. This could be due to the limited amount of data available for training. Similar to the previous case, the lack of data variation and the limitations of the architecture may have contributed to these issues.

C. DenseNet Architecture Test

Here are the results obtained from the training experiment of the DenseNet architecture for coral reef type and health level classification.

Table 9. Coral Reef Type Classification Report with DenseNet

	Precision	**Recall**	**F1 - Score**	**Accuracy**	**Training Time per Epoch**
Boulder	1.00	1.00	1.00	100%	13 second
Branching	1.00	1.00	1.00		
Table	1.00	1.00	1.00		

Table 10. Coral Reef Health Level Classification Report DenseNet

	Precision	**Recall**	**F1 - Score**	**Accuracy**	**Training Time per Epoch**
Level 1	0.92	0.85	0.88	55%	36 second
Level 2	0.50	0.92	0.65		
Level 3	0.59	0.52	0.55		
Level 4	0.36	0.48	0.41		
Level 5	0.67	0.24	0.35		
Level 6	0.56	0.50	0.53		

In the classification of coral reef types, DenseNet yielded excellent results as the model was able to predict the classes in the test data perfectly without any prediction errors. However, in the classification of coral reef health levels, the results obtained showed that the model was moderately capable of predicting the level classes. These results, although not considered good, could be attributed to the limited amount of data available and the lack of data variation.

D. VGG19 Architecture Test

Here are the results obtained from the training experiment of the VGG19 architecture for coral reef type and health level classification.

Table 11. Coral Reef Type Classification Report with VGG19

	Precision	**Recall**	**F1 - Score**	**Accuracy**	**Training Time per Epoch**
Boulder	0.69	1.00	0.82	81%	26 second
Branching	0.94	0.92	0.93		
Table	0.00	0.00	0.00		

Table 12. Coral Reef Health Level Classification Report VGG19

	Precision	**Recall**	**F1 - Score**	**Accuracy**	**Training Time per Epoch**
Level 1	0.00	0.00	0.00	24%	28 second
Level 2	0.00	0.00	0.00		
Level 3	0.24	1.00	0.39		
Level 4	0.00	0.00	0.00		
Level 5	0.00	0.00	0.00		
Level 6	0.00	0.00	0.00		

The VGG19 architecture yielded unsatisfactory results in the classification of coral reef types. It was unable to predict the "table" class, which may be due to the imbalanced data distribution compared to other classes. Similar to ResNet, the results obtained in the classification of coral reef health levels were also not good, except for level 3. This could be attributed to the limited amount of data available, as well as the lack of data variation. Additionally, the architecture used may not be sufficiently equipped to handle this particular problem.

V. Result and Interpretation

The previous experiments aimed to obtain results and provide a comparison of the MobileNetV2, DenseNet, ResNet, and VGG19 architecture models. The comparison of accuracy and the time required for training per epoch in the classification of coral reef types and health levels can be seen in the table 13.

Table 13 Comparison of Accuracy and Training Time per Epoch

Model	**Coral Reef Type Classification**		**Coral Reef Health Level Classification**	
	Accuracy	**Training Time per Epoch**	**Accuracy**	**Training Time per Epoch**
MobileNetV2	98%	5 second	26%	20 second
DenseNet	100%	13 second	55%	36 second
ResNet152	66%	12 second	22%	33 second
VGG19	81%	24 second	24%	28 second

From table 13, it can be observed that in the coral reef type classification problem, the DenseNet architecture achieved the highest accuracy with a score of 100%, followed by the MobileNetV2 architecture with an accuracy of 98%. In the health level classification problem, DenseNet also achieved the highest accuracy with a score of 55%. The difference in the best accuracy obtained is quite significant compared to the accuracies of the MobileNetV2, ResNet, and VGG19 models.

In terms of training time per epoch, the MobileNetV2 architecture showed the best results in both coral reef type and health level classification problems. This aligns with the theoretical basis that MobileNetV2 is designed to be a lightweight architecture. From the table above, the average training time per epoch for the coral reef type classification was 5 seconds, while for the health level classification, it was 20 seconds. There is a notable difference compared to the MobileNetV2, ResNet, and VGG19 models.

From the classification tests of coral reef type and health level using the MobileNetV2 model architecture, an accuracy of 98% was obtained for the classification of coral reef types, and an accuracy of 26% was obtained for the classification of health levels. Meanwhile, the average time per epoch was 5 seconds for the classification of coral reef types and 20 seconds for the classification of health levels. When looking at the classification of coral reef types, the MobileNetV2 model showed excellent results, but it performed poorly in the classification of coral reef health levels. This could be because MobileNetV2 is a model architecture suitable for tasks with low complexity. With features such as depthwise separable convolution, inverted residuals with linear bottlenecks, width multiplier, and resolution multiplier, MobileNetV2 minimizes computational load and model complexity. However, this can sacrifice performance in return. For this reason, MobileNetV2 has poor accuracy in the classification of health levels.

In the next testing using the DenseNet model architecture, an accuracy of 1.0 was obtained for the classification of coral reef types, and an accuracy of 55% was obtained for the classification of health levels. Meanwhile, the average time per epoch was 13 seconds for the classification of coral reef types and 36 seconds for the classification of health levels. Based on the accuracies, DenseNet achieved the best results in classifying coral reef types and health levels. This is because DenseNet performs well in cases with small to medium-sized datasets. DenseNet is effective in learning highly complex and abstract features. It is also suitable for cases with high computational complexity as it requires substantial computational resources. Looking at the characteristic features of DenseNet, its dense connectivity concept allows information to flow more efficiently through the model, strengthening richer and more useful representation learning. DenseNet also includes transition layers between dense blocks, which consist of 1x1 convolutions followed by pooling operations, gradually reducing spatial dimensions and model complexity. These distinctive features of DenseNet prove its reliability compared to other model architectures.

In the testing using the ResNet-152 model architecture, an accuracy of 66% was obtained for the classification of coral reef types, and an accuracy of 22% was obtained for the classification of health levels. Meanwhile, the average time per epoch was 12 seconds for the classification of coral reef types and 33 seconds for the classification of health levels. In the testing using the VGG19 model architecture, an accuracy of 81% was obtained for the classification of coral reef types, and an accuracy of 24% was obtained for the classification of health levels. Meanwhile, the average time per epoch was 24 seconds for the classification of coral reef types and 28 seconds for the classification of health levels. The results of the classification of coral reef types using the ResNet-152 and VGG19 architectures were not able to predict images belonging to the "table" class. This is because the data for the "table" class is unbalanced compared to other classes, meaning the fewer number of "table" data samples caused the model's inability to predict images belonging to the "table" class. The deep architecture of ResNet-152 and the repetitive architecture of VGG19 require a large amount of training data to create a good model. Therefore, the poor performance of the ResNet-152 and VGG19 model architectures could be attributed to their unsuitability for small datasets.

The poor accuracy results are due to the models' inability to learn patterns and features in the training data effectively. There are several things that can be done to improve these results, such as increasing the number of layers or layer width (number of filters) in the model. Adjusting training parameters, such as the learning rate, can also be helpful. If the learning rate is too low, the model may learn patterns slowly. Increasing the number of epochs and batch size, in some cases, allows the model more time to learn and train more data in each iteration, which can help overcome underfitting. Additionally, to improve accuracy, regularization techniques can be applied. Regularization is used to prevent overfitting in models. Adding regularization, such as dropout or L1/L2 regularization, to certain layers in the MobileNetV2, ResNet-152, and VGG19 architectures can help reduce underfitting.

The lack of data variation and quantity is a significant factor affecting the poor performance in the health level classification. Essentially, the health level classification of coral reefs requires more data, both in terms of quantity and variation. This directly affects the performance of the ResNet-152 and VGG19 architectures, which typically require a large amount of data. The lack of variation also impacts the results in the MobileNetV2 architecture. In this evaluation, neural network models like MobileNetV2 were unable to learn patterns and features in the training data effectively because MobileNetV2 is designed for computational efficiency and low memory usage.

VI. Conclusion

From the implementation process to the analysis of the testing results in this thesis research, several important conclusions can be drawn as follows:

- This research has produced 8 models, 4 for the classification of coral reef types and 4 for the classification of coral reef health levels.
- The trained models, namely MobileNetV2, DenseNet, ResNet152, and VGG19, can be used to predict coral reef types and health levels.
- The best design for the coral reef type classification model is achieved using the DenseNet architecture with an accuracy of 1.0, followed by the MobileNetV2 model with an accuracy of 0.98. In terms of training time, the MobileNetV2 architecture yields the best results with an average duration of 5 seconds per epoch.
- The best design for the coral reef health level classification model is obtained using the DenseNet architecture with an accuracy of 0.55. In terms of training time, the MobileNetV2 architecture yields the best results with an average duration of 20 seconds per epoch.
- The lower accuracy in the classification of coral reef health levels is attributed to the lack of data variety and quantity, which hampers the model's ability to learn patterns and features from the training data effectively.

References

[1] Jon Borbon, Jeanette Javier, Jony Llamado, Elmer Dadios, and Robert Kerwin Billones “Coral Health Identification using Image Classification and Convolutional Neural Networks” 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and

Control, Environment, and Management (HNICEM), doi : 10.1109/HNICEM54116.2021.9731905.

[2] Reef Resilience Network “Status Terumbu Karang.” [Online]. Available: https://reefresilience.org/id/stressors/reefs-are-at-risk/ [Accessed: 12- Des-2022].

[3] Ma. Shiela Angeli C. Marcos, Maricor N. Soriano and Caesar A. Saloma “Classification of coral reef images from underwater video using neural networks” National Institute of Physics, University of the Philippines, 2005, doi : 10.1364/OPEX.13.008766

[4] C.E.O. Litimco, M.G.A. Villanueva, N.G. Yecla, M.N. Soriano, and P.C. Naval, Jr. “Coral Identification Information System” 2013 IEEE International Underwater Technology Symposium (UT), doi : 10.1109/UT.2013.6519835.

[5] IBM “What is machine learning?” [Online]. Available: https://www.ibm.com/cloud/learn/machine-learning [Accessed: 13- Des-2022].

[6] Analytics Vidhya “Introduction to Convolutional Neural Networks (CNN)” [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ [Accessed: 04- Nov-2022].

[7] Great Learning “Introduction to Resnet or Residual Network” [Online]. Available: https://www.mygreatlearning.com/blog/resnet/ [Accessed: 14- Des-2022].

[8] Manufacturers Monthly “An Introduction to Deep Learning” Available: https://www.manmonthly.com.au/resources/introduction-deep-learning/ [Accessed: 14- Des-2022].

[9] Geeksforgeeks “OpenCV – Overview” [Online]. Available: https://www.geeksforgeeks.org/opencv-overview/ [Accessed: 14- Des-2022].

[10] Tensorflow “Image classification” [Online]. Available: https://www.tensorflow.org/lite/examples/image_classification/overview [Accessed: 30- Des-2022].

[11] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510-4520).

[12] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[13] Géron, A. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems." O'Reilly Media. 2019.