Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 23, No. 2, March 2024, pp. 333∼342
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v23i2.3683

Comparison of DenseNet-121 and MobileNet for Coral Reef Classification

Heru Pramono Hadi[1], Eko Hari Rachmawanto[1], Rabei Raad Ali[2]
[1]Universitas Dian Nuswantoro, Semarang, Indonesia
[2]Northern Technical University, Mosul, Iraq

Article Info

Article history:

Received January 11, 2024
Revised February 15, 2024
Accepted March 08, 2024

Keywords:

Coral Reefs
Convolutional Neural Network
Confusion Matrix
DenseNet-121
MobileNet

ABSTRACT

Coral reefs are a type of marine organism that has beauty and benefits for other sea creatures' ecosystems. However, despite its beauty and usefulness, coral reefs are vulnerable to damage such as coral bleaching, which can impact other coral reef ecosystems. This research aims to classify digital images of healthy, bleached, and dead coral reefs. This research method is DenseNet-121 and MobileNet is based on Convolutional Neural Networks. This research uses a dataset from 1582 coral reef image data with three main classes: 720 were bleached, 150 were dead, and 712 were healthy. The testing process is carried out using several forms of split datasets, namely 60:10:30, 50:10:40, and 70:10:20. The test results obtained with a data sharing percentage of 60:10:30 show that MobileNet architecture achieved 88.00% accuracy, and DenseNet-121 achieved 91.57% accuracy. Using a data split percentage of 50:10:40, MobileNet achieved 84.51% accuracy, and DenseNet- 121 achieved 90.52% accuracy. Meanwhile, with a data separation percentage of 70:10:20, MobileNet achieved 85.48% accuracy, and DenseNet-121 achieved 92.74% accuracy.

Corresponding Author:

Eko Hari Rachmawanto,+6285641016946,
Faculty of Computer Science,
Universitas Dian Nuswantoro, Semarang, Indonesia,
Email: eko.hari@dsn.dinus.ac.id

How to Cite:
H. P. Hadi, E. H. Rachmawanto, and R. R. Ali, "Comparison of DenseNet-121 and MobileNet for Coral Reef Classification", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 23, No. 2, pp. 333-342, Mar, 2024.

1. INTRODUCTION

Corals are marine invertebrates belonging to the phylum Cnidaria and the class Anthozoa [1]. Renowned for their vibrant colors and diverse patterns, they play a crucial role in the marine ecosystem [1]. Forming colonies that serve as habitats for a wide array of marine species, corals also protect coastlines from storms and erosion. However, corals are currently confronted with substantial threats stemming from climate change, pollution, and overfishing, leading to coral bleaching and mortality [2–5]. Coral reefs constitute a vast ecosystem, with estimates suggesting that their expanse accounts for less than 1% of the total area of the oceans. This ecosystem serves as the home for approximately 25% of marine species [6]. Coral reefs play a crucial role in sequestering carbon in seawater and act as natural barriers, protecting coastal areas [7].

Deep learning is a subset of Artificial Intelligence (AI) [8]. In deep learning, the concept of artificial neural networks is utilized for understanding and recognizing patterns in data [9]. This approach has found applications in various fields, including medical image analysis, natural language processing, and sentiment analysis [10–12]. Deep learning techniques have demonstrated remarkable effectiveness in tasks such as breast cancer segmentation [10] and bone age assessment [13], particularly in image analysis. Therefore, deep learning is well-suited for processing big data and handling complex tasks. One advantage of using deep learning methods is performing direct processing without preprocessing the data [14]. Convolutional Neural Network (CNN) is a component of deep learning [15]. In CNN, as it employs deep learning principles, it also incorporates the workings of artificial neural networks (ANN). Consequently, the CNN process involves several layers, including the input, hidden, and output layers [16]. The hidden layer in CNN typically consists of convolutional, pooling, and fully connected layers [17]. Each of these layers is interconnected to facilitate effective classification processes.

In this research, the classification process of healthy, dead, or bleached coral reefs will be conducted using transfer learning CNN. The objective of this study is to build a model capable of classifying coral reefs, thereby contributing to the preservation of existing coral reefs. The utilization of transfer learning in this research is motivated by CNN being a suitable method for classification, especially in digital image classification. The purpose of employing transfer learning is to leverage a pre-existing model, eliminating the need to construct a new model from scratch with a large dataset. Transfer learning is particularly advantageous when dealing with limited datasets. The chosen transfer learning CNN architectures, MobileNet and DenseNet-121, serve specific purposes. MobileNet is selected for its efficiency in image processing on devices with constrained performance, thanks to its optimal architecture [18]. On the other hand, DenseNet-121 is chosen for its strong inter-layer feature relationships, enhancing the learning of complex features and optimizing parameter usage within the architecture [19]. This makes it a good choice for real-time applications and mobile vision [18]. On the other hand, DenseNet-121 utilizes a structure that connects each layer to every subsequent layer, optimizing feature learning and improving the efficiency of parameter usage. This way, DenseNet-121 can deliver high performance with fewer parameters, making it suitable for tasks requiring detailed processing and strong feature representation [19].

In the previous research conducted by Das et al. in 2019 [20], the focus was on the classification process of brain tumors using CNN. This study aimed to construct a classification model capable of diagnosing brain tumors through CNN methods. The results obtained after testing in this research revealed a testing accuracy of 94.39%, with average precision and recall rates of 93.33% and 93%, respectively, using the developed model. In another prior study by Barbhuiya et al. in 2019 [21], the research centered on classifying sign language alphabet characters using transfer learning CNN. The objective was to build a model for classifying sign language alphabet characters and compare the effectiveness of different models in this classification process. The results showed that the testing accuracy using the CNN model with AlexNet architecture and SVM was the highest at 99.82%, compared to 99.76% with VGG16 architecture and SVM. Another research entitled ”A review of coral reef classification study using deep learning approach” by Arsat et al., 2023 [22]; Digitizing the coral reef: Machine learning of underwater spectral images enables dense taxonomic mapping of benthic habitats by Daniel et al., 2022 [23]; and ”Assessing Derawan Island’s Coral Reefs over Two Decades: A Machine Learning Classification perspective” by Masita et al, 2024 [24]. Therefore, this research aims to provide information about the performance of the CNN based on DenseNet-121 and MobileNet in the coral reef classification process. We prove that by using DenseNet-121 and MobileNet, coral reefs can be classified well. Here, we focus on visualizing DenseNet-121 and MobileNet optimal layers for a high result.

2. RESEARCH METHOD

2.1. Convolutional Neural Network (CNN)

Convolutional Neural Network is a component of deep learning [15]. In CNN, since it employs the principles of deep learning, it also incorporates the workings of artificial neural networks (ANN). Therefore, in its process, it utilizes layers. The layers in CNN include the input layer, the hidden layer typically consisting of convolution layers, pooling layers, fully connected layers, and the output layer [9]. The input layer is responsible for receiving input data. The convolution layer is tasked with feature extraction [25], and the formula for the convolution layer is provided in Equation 1.

$$(D * C)(i, j) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} D(i + x, j + y) * C(x, y) \tag{1}$$

where :
D represents the input,
C represents the kernel or filter,
i and j denote the coordinates of the convolution result between D and C,
N and M represent the dimensions of the image width and height,
- x and y are the indices for matrix iteration.

The pooling layer is designed for downsampling features, reducing their dimensions to make the previous layer's processes faster and more efficient [26], and the formula for the pooling layer is given in equation (2).

$$(Pool_{max})(i, j) = max_{x=0}^{N-1} max_{y=0}^{M-1} D(i + x, j + y) \tag{2}$$

Where:
I and j represent the positions of pixels resulting from the max pooling process.
D is the input.
N and M are the dimensions of pooling (commonly used sizes are 2x2 or 3x3).

Meanwhile, the fully connected layer is used for the classification process based on the results from the convolution and pooling layers in CNN [27], and the formula for the fully connected layer is provided in equation (3).

$$X_y = \sum_{x=1}^{N} (W_{x,y} * Actv_x) + B_y \tag{3}$$

Where:
N is the number of neurons from the previous layer.
W(x,y) is the weight connecting a neuron in the
previous layer (x) to the layer in the fully connected layer (y).
Actv(x) is the activation function in the previous layer (x).

B(y) is the bias of the fully connected layer.

Transfer learning is the process of reusing previously trained network architecture to perform new tasks [28]. The use of transfer learning is so that the previously built model can be further refined to work optimally [29]. Because the model has previously been trained in the transfer learning process, this method is suitable for use on smaller datasets so that the process can be more optimal [30]. Aneja et al. [31] say that this transfer learning technique is very helpful for building models when we do not have enough datasets to train the model. In this research, the CNN transfer learning method will be used, namely MobileNet and DenseNet-121 [32], where this model has been trained using imagiNet data so that the model built later can implement the deep network method [33].

```
dataset = load_dataset #using for load the dataset that would be used
train_data, validation_data, test_data = split_dataset(dataset) #split the dataset for
train, valid and test
pretrained_model = load_pretrained_model()
for layer in pretrained_model.layers:
    layer.trainable = False
model = custom_layer(pretrained_model)
model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
history = model.fit(train_data, epochs=10, validation_data=validation_data)
```

2.2. Proposed Method

In this study, the classification process will be conducted to determine whether a coral reef is healthy, bleached, or dead, aiming to provide valuable insights into the state of the marine environment. Through this process, steps can be taken to preserve the existing coral reef ecosystem by implementing targeted conservation efforts based on the identified conditions. The classification workflow, which encompasses the analysis of marine images and the application of advanced machine learning techniques, is illustrated in Figure 1, outlining the systematic approach employed to assess and categorize coral reefs, thus contributing to the ongoing conservation and management of these vital marine ecosystems.

An explanation of the steps in the classification process is provided in Figure 1. The image data reading process involves reading the coral reef image dataset obtained, typically from sources like kaggle.com. This dataset serves as the foundation for the classification process. The Data Division Process involves the image data is then divided into three main parts: training data (60%), validation data (10%), and testing data (30%). This division is crucial for effectively training, validating, and testing the model. The classification model utilizes transfer learning CNN with MobileNet and DenseNet-121 architectures. Transfer learning leverages previously trained models to enhance performance on a smaller dataset. The layers used, such as convolutional layers, pooling layers, and fully connected layers, are defined for each architecture. Details of these layers can be seen in Table 1 and Table 2. After the initialization process and creating the layers to be used, the next step is to perform data augmentation. This involves rotation by 40 degrees, width shift by 0.25, height shift by 0.2, shear by 0.2, zoom by 0.1, horizontal flip (set to true), and using the nearest fill mode. Once the data and the model are prepared, the training process will be conducted concurrently with validation to ensure that the model does not experience overfitting. Subsequently, the trained model will undergo a testing process. Following the model testing process, the model's performance will be evaluated by calculating the confusion matrix. This matrix provides insights into the model's performance in the classification process.

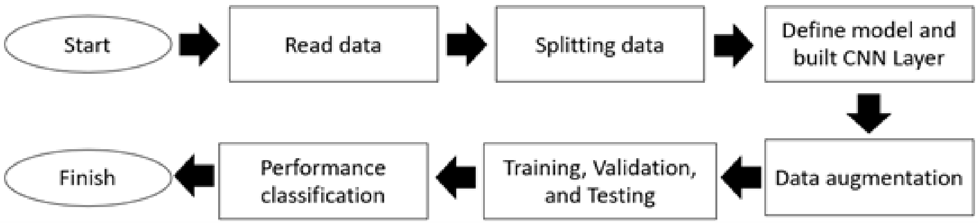

Figure 1. Classification Process

Table 1. MobileNet Proposed Layers

Layer (Type)	Output Shape	Param #
Input_1 (InputLayer)	[(None, 100, 100, 3)]	0
Conv1(Conv2D)	(None, 50, 50, 32)	864
Conv1_bn(BatchNormalization)	(None, 50, 50, 32)	128
Conv1_relu(ReLU)	(None, 50, 50, 32)	0
...		
...MobileNetV2Layers ...		
Global_average_pooling2d	(GI(None, 1280)	0
Dense(Dense)	(None, 512)	655872
Dense_1(Dense)	(None, 3)	1539
Total params : 2,386,147		
Trainable params : 164,355		
Non-trainable params : 2,221,792		

Table 2. DenseNet-121 Proposed Layers

Layer (Type)	Output Shape	Param #
DenseNet-121 (Funtional)	(None, 7, 7, 1024)	7037504
Global_average_pooling2d_4 (GlobalAveragePooling2D)	(None, 1024)	0
Fkatten_3 (Flatten)	(None, 1024)	0
Dense_9 (Dense)	(None, 256)	262400
Dense_10 (Dense)	(None, 512)	131584
Dense_11 (Dense)	(None, 512)	262656
Dense_12 (Dense)	(None, 3)	1539
Total params : 7,695,683		
Trainable params : 658,179		
Non-trainable params : 7,037,504		

2.3. Data Visualization

In this research, the classification process will utilize a coral reef dataset obtained from kaggle.com, which is a public dataset. The dataset used in this study consists of a total of 1582 coral reef image data, further divided into 3 classes: healthy, bleached, and dead. The distribution of data per class includes 712 healthy data, 720 bleached data, and 150 dead data. The visualization of the dataset used is provided in Figure 2. Out of the total 1582 data, it will be divided into training data, validation data, and testing data, with data split percentages of 60:10:30, 50:10:40, and 70:10:20. The purpose of using different data split ratios is to observe which data split ratio produces the most optimal testing accuracy. The training data enables the model to learn and recognize patterns in the data. Validation data is employed to check the model and ensure that the constructed model does not suffer from overfitting. Meanwhile, testing data is crucial for evaluating the model's performance in the classification process. Preprocessing will be performed through data augmentation using the dataset used. According to Chlap et al. [34], in the augmentation process, the image undergoes changes in shape and position to manipulate the image intensity values, producing an augmented image. This helps minimize the occurrence of overfitting. The data augmentation process includes rotation by 40 degrees, width shift by 0.25, height shift by 0.2, shear by 0.2, zoom by 0.1, horizontal flip set to true, and fill mode using the nearest method. The purpose of conducting data manipulation through data augmentation or variations in data split is to enable the built model to achieve optimal testing results, ideally reaching 100% [23].

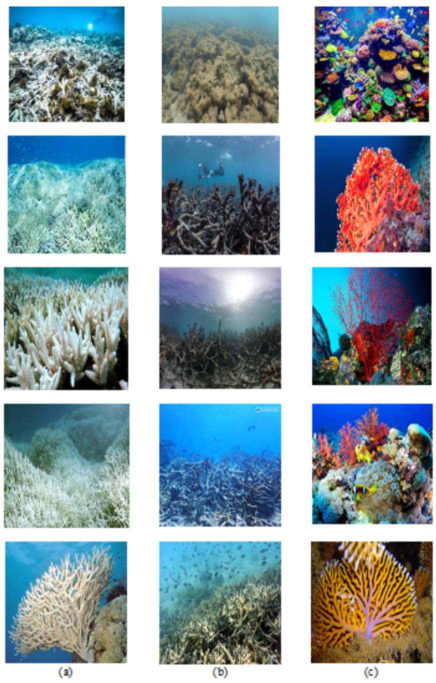

Figure 2. Visualization of a sample of datasets: (a) healthy coral, (b) bleached coral, (c) dead coral

2.4. Testing Method

A confusion matrix is a matrix used to calculate the performance of a model. The confusion matrix has four values: True Positive, True Negative, False Positive, and False Negative. Performance metrics such as precision, recall, and F1-score can be calculated from these four values. The formulas for calculating these performance metrics are shown in (4) for the precision formula, (5) for the recall formula, and (6) for the f-1 score formula.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

3. RESULT AND ANALYSIS

The chosen models for this task are pre-trained transfer learning CNN models, specifically MobileNet and DenseNet-121, previously trained on the ImageNet dataset. MobileNet is chosen due to its known lightweight and efficient design, making it suitable for implementation on mobile or small devices with limited resources. Therefore, this research will discuss the classification process for this dataset, providing an overview of how the classification is conducted on this specific dataset. Once the dataset and models are prepared, the next step involves the training process on the dataset. After completing the training process for the models, the training and validation results are obtained, as presented in Table 3.

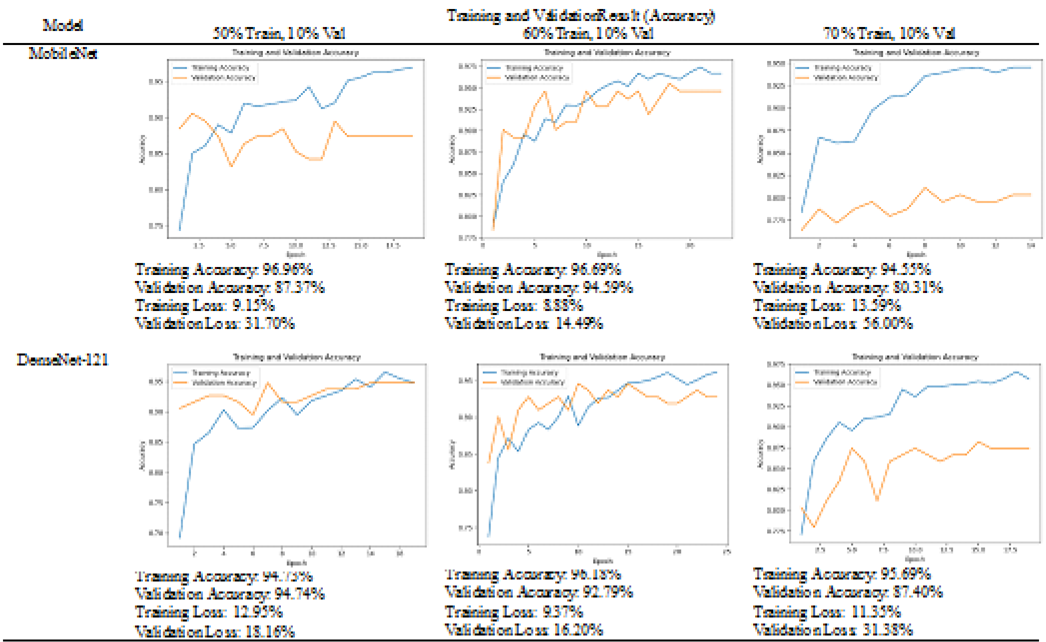

Figure 3. A Comparison Result of Training and Validation Results Using MobileNet Versus DenseNet-121

Table 13 displays the training results of the constructed MobileNet and DenseNet-121 models using different data split variations. From Table 1, it can be observed that the models trained with data split variations of 50:10 (50% Train, 10% Val) and 60:10 (60% Train, 10% Val) did not experience overfitting, as indicated by the model's graph showing no significant gap. However, overfitting occurred when the model was trained with a data split of 70:10 (70% Train, 10% Val), as evidenced by the gap between training and validation results on the graph. This overfitting may happen because the model fails to generalize data it has not seen before (test data) and can only recognize the data used during the training process. Table 1 also shows that the MobileNet model achieved the best training results without overfitting when using a 60:10 data split (60% Train, 10% Val), with a testing accuracy of 96.69% and a validation accuracy of 94.59%. On the other hand, the DenseNet-121 model achieved the highest training and validation accuracy when using a 50:10 data split (50% Train, 10% Val), with values of 94.73% and 94.74%, respectively. The best results are considered to be those with training and validation values that do not have a significant gapfrom these results, had been seen that the MobileNet model, with a 60:10 data split, achieved the most optimal testing results with training accuracy at 96.69% and validation accuracy at 94.59%. This indicates that the MobileNet model with a 60:10 data split can effectively recognize patterns in the image data used, and it is expected to yield optimal results in the classification process as well. After the training and validation processes, the next step is testing the previously trained and validated models. The results of the model testing are presented in Table 3.

Table 3. Testing Result

Model	**Split Data**	**Testing Result**				
		Accuracy	**Precision**	**Recall**	**F1-Score**	**Support**
MobileNet	50:10:40	84.51%	75.00%	64.00%	64.00%	633
	60:10:30	88.00%	77.00%	73.00%	74.00%	475
	70:10:20	85.48%	74.00%	69.00%	70.00%	317
DenseNet-121	50:10:40	90.52%	87.00%	76.00%	79.00%	633
	60:10:30	91.57%	89.00%	76.00%	80.00%	475
	70:10:20	92.74%	91.00%	80.00%	83.00%	317

Table 3 shows the results of the model testing process that was previously trained using various data split schemes. It can be observed that the MobileNet model achieved the best testing accuracy, precision, recall, and f1-score when using a data split scheme of 60% training, 10% validation, and 30% testing. The values obtained were an accuracy of 88.00%, precision of 77.00%, recall of 73.00%, and f1-score of 74.00%. These results indicate that the MobileNet model built with a 60:10:30 data split accurately predicts and can correctly classify each class while maintaining a good balance between precision and recall. On the other hand, the DenseNet-121 model showed optimal performance when the data split scheme was 70% training, 10% validation, and 20% testing. This configuration achieved a testing accuracy of 92.74%, precision of 91.00%, recall of 80.00%, and f1-score of 83.00%. These values suggest that the DenseNet-121 model has excellent predictive accuracy and a good balance of metrics. Comparing the best values obtained for both models, it can be seen that the DenseNet-121 model outperforms the MobileNet model in the coral reef classification process. The difference in testing accuracy between the two models is 4.74%. This difference indicates that the DenseNet-121 model performs better than the previously built MobileNet model.

In this study, the classification process did not achieve 100% accuracy, primarily due to the nature of the dataset. The dataset used is semi-realistic and collected from photographs taken by divers. This introduces the possibility of very similar data points across different classes, which can impact the model's accuracy. Additionally, this study's manual or ”Rules of Thumb” parameter selection may have limitations in optimizing results. Therefore, due to these factors, the testing accuracy obtained in this research has not reached its maximum potential. However, considering the given constraints and challenges, the study has achieved the most optimal results.

4. CONCLUSION

After testing the MobileNet and DenseNet-121 models trained with various data split variations, the best results were obtained for MobileNet with a data split scheme of 60% training, 10% validation, and 30% testing, achieving a testing accuracy of 88.00%. Meanwhile, DenseNet-121, with a data split scheme of 70% training, 10% validation, and 20% testing, achieved the highest testing accuracy at 92.74%. Based on these results, it can be concluded that the DenseNet-121 model, built with a data split of 70% training, 10% validation, and 20% testing, performs the classification process more optimally than the MobileNet model or other data split schemes. Thus, it can be concluded that the constructed DenseNet-121 model provides more optimal performance.

For future research, it is recommended to conduct testing using other CNN architectures such as VGG16, VGG19, ResNet, GoogleNet, etc. Additionally, future studies are encouraged to explore ensemble learning techniques or the combination of multiple classification models. Furthermore, the optimization of hyperparameters, such as Particle Swarm Optimization, Genetic Algorithm, and others, could be considered in subsequent research.

5. ACKNOWLEDGEMENTS

The Acknowledgments section is optional. Research sources can be included in this section.

6. DECLARATIONS

AUTHOR CONTIBUTION

This research has been supported by the University of Dian Nuswantoro based on Research Grant No. 109/A.38-04/UDN-09/XI/2023 as one of the output research.

FUNDING STATEMENT

COMPETING INTEREST

REFERENCES

[1] E. Bayraktarov, A. T. Banaszak, P. Montoya Maya, J. Kleypas, J. E. Arias-Gonzlez, M. Blanco, J. Calle-Trivio, N. Charuvi, C. Corts-Useche, V. Galvn, M. A. Garca Salgado, M. Gnecco, S. D. Guendulain-Garca, E. A. Hernndez Delgado, J. A. Marn Moraga, M. F. Maya, S. Mendoza Quiroz, S. Mercado Cervantes, M. Morikawa, G. Nava, V. Pizarro, R. I. Sellares-Blasco, S. E. Suleimn Ramos, T. Villalobos Cubero, M. F. Villalpando, and S. Fras-Torres, "Coral reef restoration efforts in Latin American countries and territories," *PLOS ONE*, vol. 15, no. 8, p. e0228477, Aug. 2020. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0228477

[2] T. P. Hughes, J. T. Kerry, A. H. Baird, S. R. Connolly, T. J. Chase, A. Dietzel, T. Hill, A. S. Hoey, M. O. Hoogenboom, M. Jacobson, A. Kerswell, J. S. Madin, A. Mieog, A. S. Paley, M. S. Pratchett, G. Torda, and R. M. Woods, "Global warming impairs stockrecruitment dynamics of corals," *Nature*, vol. 568, no. 7752, pp. 387–390, Apr. 2019, https://doi.org/10.1038/s41586-019-1081-y. [Online]. Available: https://www.nature.com/articles/s41586-019-1081-y

[3] L. B. DeFilippo, L. C. McManus, D. E. Schindler, M. L. Pinsky, M. A. Colton, H. E. Fox, E. W. Tekwa, S. R. Palumbi, T. E. Essington, and M. M. Webster, "Assessing the potential for demographic restoration and assisted evolution to build climate resilience in coral reefs," *Ecological Applications*, vol. 32, no. 7, p. e2650, Oct. 2022. [Online]. Available: https://esajournals.onlinelibrary.wiley.com/doi/10.1002/eap.2650

[4] M. Slattery, M. S. Pankey, and M. P. Lesser, "Annual Thermal Stress Increases a Soft Corals Susceptibility to Bleaching," *Scientific Reports*, vol. 9, no. 1, p. 8064, May 2019, https://doi.org/10.1038/s41598-019-44566-9. [Online]. Available: https://www.nature.com/articles/s41598-019-44566-9

[5] T. A. Courtney, B. B. Barnes, I. Chollett, R. Elahi, K. Gross, J. R. Guest, I. B. Kuffner, E. A. Lenz, H. R. Nelson, C. S. Rogers, L. T. Toth, and A. J. Andersson, "Disturbances drive changes in coral community assemblages and coral calcification capacity," *Ecosphere*, vol. 11, no. 4, p. e03066, Apr. 2020, https://doi.org/10.1002/ecs2.3066. [Online]. Available: https://esajournals.onlinelibrary.wiley.com/doi/10.1002/ecs2.3066

[6] A. Raphael, Z. Dubinsky, D. Iluz, and N. S. Netanyahu, "Neural Network Recognition of Marine Benthos and Corals," *Diversity*, vol. 12, no. 1, p. 29, Jan. 2020, https://doi.org/10.3390/d12010029. [Online]. Available: https://www.mdpi.com/1424-2818/12/1/29

[7] M. Asha Paul, P. Arockia Jansi Rani, and J. Liba Manopriya, "Gradient Based Aura Feature Extraction for Coral Reef Classification," *Wireless Personal Communications*, vol. 114, no. 1, pp. 149–166, Sep. 2020, https://doi.org/10.1007/s11277-020-07355-6. [Online]. Available: https://link.springer.com/10.1007/s11277-020-07355-6

[8] L. Von Chamier, R. F. Laine, J. Jukkala, C. Spahn, D. Krentzel, E. Nehme, M. Lerche, S. Hernndez-Prez, P. K. Mattila, E. Karinou, S. Holden, A. C. Solak, A. Krull, T.-O. Buchholz, M. L. Jones, L. A. Royer, C. Leterrier, Y. Shechtman, F. Jug, M. Heilemann, G. Jacquemet, and R. Henriques, "Democratising deep learning for microscopy with ZeroCostDL4Mic," *Nature Communications*, vol. 12, no. 1, p. 2276, Apr. 2021, https://doi.org/10.1038/s41467-021-22518-0. [Online]. Available: https://www.nature.com/articles/s41467-021-22518-0

[9] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognition Letters*, vol. 141, pp. 61–67, Jan. 2021, https://doi.org/10.1016/j.patrec.2020.07.042. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167865520302981

[10] Agus Perdana Windarto, Anjar Wanto, S Solikhun, and Ronal Watrianthos, "A Comprehensive Bibliometric Analysis of Deep Learning Techniques for Breast Cancer Segmentation: Trends and Topic Exploration (2019-2023)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 5, pp. 1155–1164, Oct. 2023, https://doi.org/10.29207/resti.v7i5.5274. [Online]. Available: http://jurnal.iaii.or.id/index.php/RESTI/article/view/5274

[11] J. Hu, X. Deng, Y. Pan, Y. Wang, and W. Jin, "Temporal Encoded Deep Learning Radiomics Model for Preoperative Prediction of Microvascular Invasion in Hepatocellular Carcinoma," *Journal of Medical and Biological Engineering*, vol. 43, no. 5, pp. 623–632, Oct. 2023, https://doi.org/10.1007/s40846-023-00829-5. [Online]. Available: https://link.springer.com/10.1007/s40846-023-00829-5

[12] K. H. Manguri, R. N. Ramadhan, and P. R. Mohammed Amin, "Twitter Sentiment Analysis on Worldwide COVID-19 Outbreaks," *Kurdistan Journal of Applied Research*, pp. 54–65, May 2020, https://doi.org/10.24017/covid.8. [Online]. Available: https://www.spu.edu.iq/kjar/index.php/kjar/article/view/512

[13] P. H. Kim, H. M. Yoon, J. R. Kim, J.-Y. Hwang, J.-H. Choi, J. Hwang, J. Lee, J. Sung, K.-H. Jung, B. Bae, A. Y. Jung, Y. A. Cho, W. H. Shim, B. Bak, and J. S. Lee, "Bone Age Assessment Using Artificial Intelligence in Korean Pediatric Population: A Comparison of Deep-Learning Models Trained With Healthy Chronological and Greulich-Pyle Ages as Labels," *Korean Journal of Radiology*, vol. 24, no. 11, p. 1151, 2023, https://doi.org/10.3348/kjr.2023.0092. [Online]. Available: https://kjronline.org/DOIx.php?id=10.3348/kjr.2023.0092

[14] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher, "Deep learning-enabled medical computer vision," *npj Digital Medicine*, vol. 4, no. 1, p. 5, Jan. 2021, https://doi.org/10.1038/s41746-020-00376-2. [Online]. Available: https://www.nature.com/articles/s41746-020-00376-2

[15] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evolutionary Intelligence*, vol. 15, no. 1, pp. 1–22, Mar. 2022, https://doi.org/10.1007/s12065-020-00540-3. [Online]. Available: https://link.springer.com/10.1007/s12065-020-00540-3

[16] M. B. Bora, D. Daimary, K. Amitab, and D. Kandar, "Handwritten Character Recognition from Images using CNN-ECOC," *Procedia Computer Science*, vol. 167, pp. 2403–2409, 2020, https://doi.org/10.1016/j.procs.2020.03.293. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050920307596

[17] A. inar and M. Yildirim, "Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture," *Medical Hypotheses*, vol. 139, p. 109684, Jun. 2020, https://doi.org/10.1016/j.mehy.2020.109684. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306987720301717

[18] J. Chen, D. Zhang, M. Suzauddola, Y. A. Nanehkaran, and Y. Sun, "Identification of plant disease images via a squeezeandexcitation MobileNet model and twice transfer learning," *IET Image Processing*, vol. 15, no. 5, pp. 1115–1127, Apr. 2021, https://doi.org/10.1049/ipr2.12090. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1049/ipr2.12090

[19] Y.-D. Zhang, S. C. Satapathy, X. Zhang, and S.-H. Wang, "COVID-19 Diagnosis via DenseNet and Optimization of Transfer Learning Setting," *Cognitive Computation*, Jan. 2021, https://doi.org/10.1007/s12559-020-09776-8. [Online]. Available: http://link.springer.com/10.1007/s12559-020-09776-8

[20] S. Das, O. F. M. R. R. Aranya, and N. N. Labiba, "Brain Tumor Classification Using Convolutional Neural Network," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*. Dhaka, Bangladesh: IEEE, May 2019, pp. 1–5, https://doi.org/10.1109/ICASERT.2019.8934603. [Online]. Available: https://ieeexplore.ieee.org/document/8934603/

[21] A. A. Barbhuiya, R. K. Karsh, and R. Jain, "CNN based feature extraction and classification for sign language," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 3051–3069, Jan. 2021, https://doi.org/10.1007/s11042-020-09829-y. [Online]. Available: https://link.springer.com/10.1007/s11042-020-09829-y

[22] T. N. T. Arsad, E. A. Awalludin, Z. Bachok, W. N. J. H. W. Yussof, and M. S. Hitam, "A review of coral reef classification study using deep learning approach," Kuala Terengganu, Malaysia, 2023, p. 050005, https://doi.org/10.1063/5.0110245. [Online]. Available: https://pubs.aip.org/aip/acp/article/2879531

[23] D. Schrholz and A. Chennu, "Digitizing the coral reef: Machine learning of underwater spectral images enables dense taxonomic mapping of benthic habitats," *Methods in Ecology and Evolution*, vol. 14, no. 2, pp. 596–613, Feb. 2023, https://doi.org/10.1111/2041-210X.14029. [Online]. Available: https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.14029

[24] M. D. M. Manessa, M. A. F. Ummam, A. F. Efriana, J. M. Semedi, and F. Ayu, "Assessing Derawan Islands Coral Reefs over Two Decades: A Machine Learning Classification Perspective," *Sensors*, vol. 24, no. 2, p. 466, Jan. 2024, https://doi.org/10.3390/s24020466. [Online]. Available: https://www.mdpi.com/1424-8220/24/2/466

[25] S. Sakib, N. Ahmed, A. J. Kabir, and H. Ahmed, "An Overview of Convolutional Neural Network: Its Architecture and Applications," MATHEMATICS & COMPUTER SCIENCE, preprint, Nov. 2018, https://doi.org/10.20944/preprints201811.0546.v1. [Online]. Available: http://www.preprints.org/manuscript/201811.0546/v1

[26] H. Gholamalinezhad and H. Khosravi, "Pooling Methods in Deep Neural Networks, a Review," 2020, https://doi.org/10.48550/ARXIV.2009.07485. [Online]. Available: https://arxiv.org/abs/2009.07485

[27] X. Y. Wu, "A hand gesture recognition algorithm based on DC-CNN," *Multimedia Tools and Applications*, vol. 79, no. 13-14, pp. 9193–9205, Apr. 2020, https://doi.org/10.1007/s11042-019-7193-4. [Online]. Available: http://link.springer.com/10.1007/s11042-019-7193-4

[28] S. M. Hassan, A. K. Maji, M. Jasiski, Z. Leonowicz, and E. Jasiska, "Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach," *Electronics*, vol. 10, no. 12, p. 1388, Jun. 2021, https://doi.org/10.3390/electronics10121388. [Online]. Available: https://www.mdpi.com/2079-9292/10/12/1388

[29] K. Thenmozhi and U. Srinivasulu Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," *Computers and Electronics in Agriculture*, vol. 164, p. 104906, Sep. 2019, https://doi.org/10.1016/j.compag.2019.104906. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0168169919310695

[30] T. Rahman, M. E. H. Chowdhury, A. Khandakar, K. R. Islam, K. F. Islam, Z. B. Mahbub, M. A. Kadir, and S. Kashem, "Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using Chest X-ray," *Applied Sciences*, vol. 10, no. 9, p. 3233, May 2020, https://doi.org/10.3390/app10093233. [Online]. Available: https://www.mdpi.com/2076-3417/10/9/3233

[31] N. Aneja and S. Aneja, "Transfer Learning using CNN for Handwritten Devanagari Character Recognition," in *2019 1st International Conference on Advances in Information Technology (ICAIT)*. Chikmagalur, India: IEEE, Jul. 2019, pp. 293–296, https://doi.org/10.1109/ICAIT47043.2019.8987286. [Online]. Available: https://ieeexplore.ieee.org/document/8987286/

[32] A. Susanto, C. A. Sari, E. H. Rachmawanto, I. U. W. Mulyono, and N. M. Yaacob, "A Comparative Study of Javanese Script Classification with GoogleNet, DenseNet, ResNet, VGG16 and VGG19," *Scientific Journal of Informatics*, vol. 11, no. 1, pp. 31–40, Jan. 2024, https://doi.org/10.15294/sji.v11i1.47305. [Online]. Available: https://journal.unnes.ac.id/nju/sji/article/view/47305

[33] K. M. Hosny, M. A. Kassem, and M. M. Foaud, "Classification of skin lesions using transfer learning and augmentation with Alex-net," *PLOS ONE*, vol. 14, no. 5, p. e0217293, May 2019, https://doi.org/10.1371/journal.pone.0217293. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0217293

[34] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no. 5, pp. 545–563, Aug. 2021, https://doi.org/10.1111/1754-9485.13261. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/1754-9485.13261