Indian Journal of Geo Marine Sciences
Vol. 53 (12), December 2024, pp. 764-775
DOI: 10.56042/ijms.v53i12.13956

Research Article

CORAL–Classification of Reefs and Analysis using Learning algorithms of image processing

J Bhagat[a], M Tandel[b] & G Saha*[,c]

[a]Department of Information Technology, Dharamsinh Desai Institute of Technology, Dharamsinh Desai University, Chalali, Nadiad, Gujarat 387 003, India

[b]Department of Machine Learning in Sciences, University of Nottingham, UK

[c]Department of Electronics and Communications, G H Patel college of Engineering and Technology, The Charutar Vidya Mandal (CVM) University, Vallabh Vidyanagar, Gujarat – 388 120, India

*[E-mail: geetalisaha@gcet.ac.in]

Received 25 September 2024; revised 04 December 2024

Coral bleaching, driven primarily by rising sea temperatures, poses a severe threat to coral reefs and the millions who depend on these resources. This study investigates the potential of deep learning for automated bleaching detection, a crucial step towards effective monitoring and conservation. Study evaluated five classification algorithms, each paired with three feature extractors, using a publicly available dataset of 1150 coral images. The findings demonstrate the superior performance of the DenseNet-Logistic Regression model, achieving the highest average accuracy (83 %), F1-score (0.84), and precision (0.85), highlighting its effectiveness in capturing subtle bleaching indicators. While this research underscores the promising outcomes of deep learning for this critical task, further investigation with larger, more diverse datasets is warranted to develop highly accurate and generalisable models for safeguarding these vital ecosystems.

[**Keywords**: Biodiversity, Coral reef, Coral bleaching, Deep learning algorithm, Image processing, India]

Introduction

Coral reefs are among the most diverse and productive ecosystems on the planet, providing food, income, and coastal protection for millions of people worldwide. Though they cover less than 0.1 % of the Earth's surface, these bustling ecosystems are essential to the health of our oceans. Coral reefs, often called the "rainforests of the sea", provide a home for a quarter of all marine species and play a critical role in maintaining water quality[1].

These coral ecosystems, made up of tiny living organisms called polyps, form a foundation upon which countless marine species thrive. The corals themselves build intricate and sprawling reef structures over thousands of years, creating an underwater tapestry that supports a complex web of life. These reefs cater to around 4,000 fish species, 840 coral species, and over 1 million other organisms[2]. However, despite their beauty and importance, coral reefs are facing a silent crisis that threatens their existence: coral bleaching. Corals are known for their vibrant colours, but they lose these colours and resemble a white skeleton when bleached. This is the stark reality of a bleached coral, stripped of its life-giving symbiotic algae, known as zooxanthellae. These microscopic algae provide corals with their vibrant colours and, crucially, their primary source of food. The relationship between corals and zooxanthellae is a delicate symbiosis, where the algae photosynthesise and produce nutrients that feed the coral polyps, while the corals offer the algae a protected environment and access to sunlight.

When corals are stressed by environmental factors, such as elevated sea surface temperatures, they expel the zooxanthellae, leading to their pale white appearance and putting them at a higher risk of disease and death[3]. Research indicates that coral bleaching typically occurs when sea surface temperatures exceed the local summer maximum by 1 °C (1.8 °F) or more for an extended period[4]. The Degree Heating Week (DHW) metric is used to quantify heat stress, measuring how much and for how long temperatures exceed this bleaching threshold. Studies have shown that at 4 °C-weeks, significant coral bleaching is likely. At 8 °C-weeks or higher, severe bleaching and increased coral mortality are expected[5-7].

The primary reason behind this devastating phenomenon is rising sea temperatures. Global warming, driven by the accumulation of greenhouse gases in the atmosphere, has led to an increase in ocean temperatures. Even a slight rise in temperature can cause thermal stress in corals, triggering the bleaching process. In recent decades, mass bleaching events have become more frequent and severe, with some of the most significant episodes occurring in 1998, 2010, and 2016. During these events, vast stretches of coral reefs around the world were left bleached[8]. Based on the 2022 IPCC Special Report[9], Table 1 maps the relationship between temperature rise and percentage of coral reef impacts. Numerous studies have demonstrated the devastating impact of different environmental phenomena on coral reef ecosystems, as represented below.

The research focusing on coral species' susceptibility to bleaching emphasised the need for a simple comparison method. Coral fragments from five species in southern Taiwan were placed in a shared aquarium with varying temperatures. Measurements of Relative Grayscale (RG %) at different stages were used to create a Bleaching Time Index (BTI), which ranked the susceptibility of the coral species[10]. Analysis of Sea Surface Temperature (SST) and coral bleaching using MODIS-Aqua and MODIS-Terra data showed a strong correlation between SST and bleaching in 2016, with no significant differences between the satellite sources[11]. A novel method was introduced to identify coral bleaching-prone areas using Marine Heatwaves (MHWs) in the Red Sea, tuning MHW parameters to detect known bleaching regions, revealing an emerging pattern of extreme temperature changes[12]. Coral bleaching in the Gulf of Kachchh in 2016 was linked to prolonged positive temperature anomalies, resulting in significant bleaching events[13]. Total organic carbon levels in the coral reef ecosystem around Agatti Island, Lakshadweep, were studied, stressing the importance of understanding organic carbon dynamics in these ecosystems.[14]

In this paper, the efforts are laid (irrespective of coral species classification) to identify bleached and un-bleached corals through advanced image processing techniques. Study approach involved several key steps. First, pre-processing of the coral images to enhance their quality and ensure they were suitable for analysis. Next, is the employment of various deep learning algorithms to extract features from these pre-processed images. These algorithms are designed to automatically identify and learn important patterns and characteristics from the images. Following feature extraction, multiple classification algorithms were applied to the extracted features. These classifiers were trained to differentiate between bleached and non-bleached corals based on the learned patterns. This study demonstrates the potential of combining deep learning feature extraction with classification techniques to address the pressing issue of coral bleaching, offering a promising tool for monitoring and protecting coral reef ecosystems.

Image processing plays a wide and critical role in several scientific applications, enabling the extraction of valuable information from various forms of visual data. Some of the instances of data extraction using image processing for scientific applications include classification of mangrove species using UAV data, wherein, the machine learning algorithms process visible spectrum images through segmentation and feature extraction[15]; land-use changes in mangroves and assess the water quality in Malad Creek, Mumbai using remote sensing methods, including image classification and change detection[16-17]. In coral reef studies, underwater images were processed with segmentation and enhancement techniques to detect coral diseases[18]. Automated mineral identification in ferromanganese encrustations involve Scanning Electron Microscopy (SEM) images, using Back Scattered Electron (BSE) imaging and Energy Dispersive X-ray Spectroscopy (EDS)[19]. Additionally, oil spill detection in marine environments employs Convolutional Neural Networks (CNNs) for image classification, providing high accuracy in identifying and monitoring oil spills from satellite images[20]. Further, satellite images are also processed and used for sea surface temperature prediction, utilising data fusion and pre-processing methods to assess temperature variations[21].

Table 1 — Data on temperature in degree and percentage of bleaching impact

Temperature rise (°C)	Coral reef impact	Source
1.0	70 – 90 % of coral reefs at risk of long-term degradation	Impacts of 1.5 °C global warming on natural and human systems[10]
1.5	> 90 % of coral reefs at risk of severe degradation	
2.0	> 99 % of coral reefs expected to experience severe bleaching events	

Materials and Methods

This study employs a publicly available dataset of coral images, curated explicitly for research on bleached coral detection. The dataset, from Kaggle[22] contains 1,582 images captured primarily by underwater drones from the Great Barrier Reef of Australia. For this analysis, dead coral images were excluded, and a total of 1,150 images were selected, divided into two distinct classes and are used to facilitate the binary classification of coral bleaching:

- Bleached corals: 580 images
- Unbleached corals: 570 images

This dataset was originally prepared by Sonain Jamil *et al.*[23] and used in their article, *"Bag of Features (BoF) Based Deep Learning Framework for Bleached Corals Detection"*, serving as a resource for coral classification research. This balanced dataset provides a foundation for training and evaluating deep learning models aimed at automating the identification of bleached corals from underwater images. The sample images from the dataset are shown in Figure 1.

The process of image classification through transfer learning typically involves utilising a pre-trained neural network to extract features and classify images. Transfer learning proves beneficial when dealing with limited labelled data for a specific task, allowing the utilization of knowledge acquired from training on a larger dataset for a related task. In this case, the objective is to categorise images as either bleached or non-bleached. The approach in current study involves the use of deep learning algorithms in conjunction with machine learning classifiers for precise identification of bleaching events. This strategy not only enhances accuracy but also improves the feature extraction process. The entire process is depicted in the flowchart (Fig. 2).

Coral image pre-processing

I. Resizing

Let the original coral image be represented by a matrix I of dimensions $H \times W \times C$, where, H is height, W is width, and C is the number of colour channels. Resizing it to 128 × 128 gives a new matrix I'.

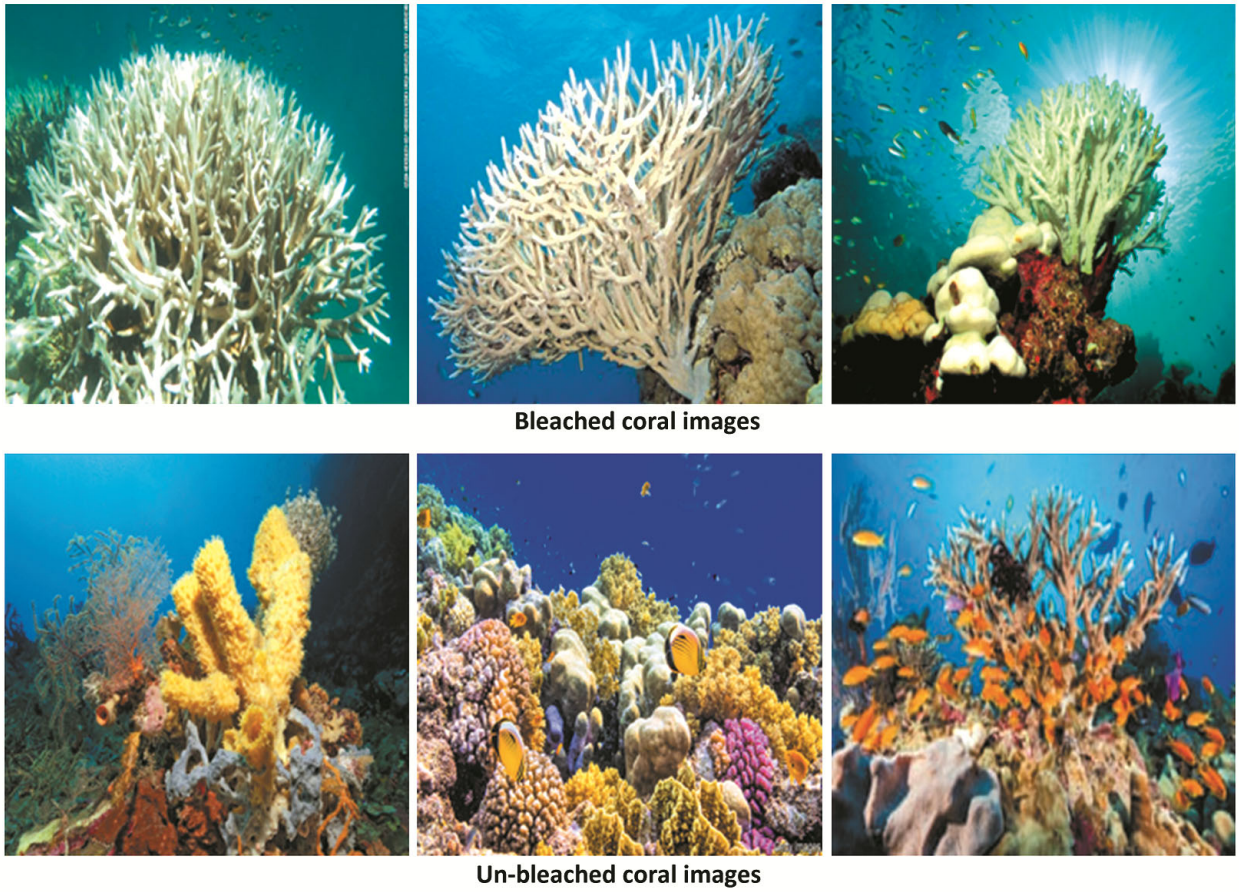

Fig. 1 — Healthy and bleached coral images of the Great Barrier Reef of Australia from dataset

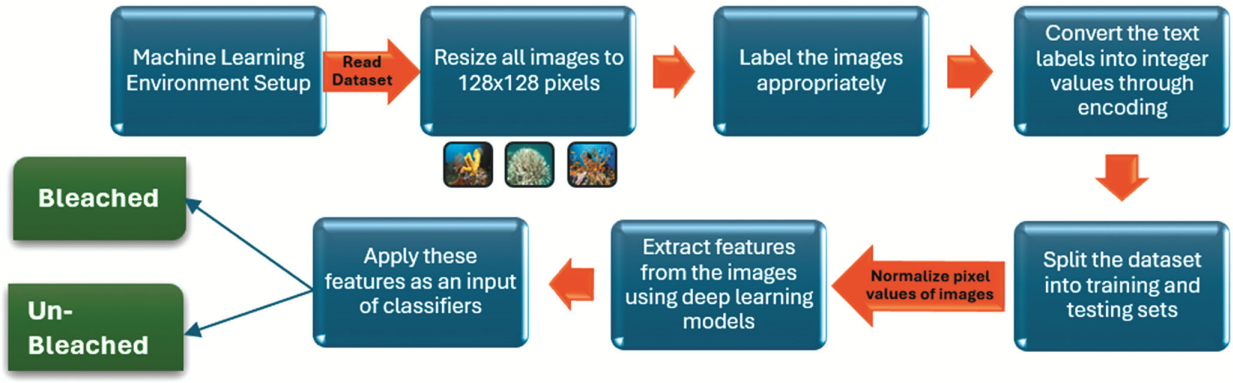

Fig. 2 — Flowchart explaining research methodology

II. Normalisation

Normalisation of pixel values typically using min-max normalisation:

$$I'_{norm}(i,j,c) = \frac{I'(i,j,c) - (I')}{\max(I') - \min(I')}$$

Where, *i, j, c* represents the pixel value at location *i, j* in channel *c*.

Feature extraction using transfer learning

Three pre-trained deep learning models were employed for feature extraction:

I. DenseNet

DenseNet introduces dense connections between layers, where each layer receives input from all previous layers and passes its own feature maps to all subsequent layers. This helps in mitigating the vanishing gradient problem and improving feature reuse. Let $[x_0, x_1, \ldots, x_{l-1}]$ be the feature maps learned by the previous layers up to layer *l*. The *l*-th layer receives input from all preceding layers:

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}])$$

Here, $H_l(\cdot)$ represents a non-linear transformation, which may include operations like batch normalisation, ReLU activation, and convolution. The concatenation of feature maps from all previous layers is represented by $[x_0, x_1, \ldots, x_{l-1}]$.

The growth rate k controls the number of filters in each layer. For the *l*-th layer:

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]) \quad \text{with} \quad x_l \in R^{H \times W \times k}$$

Where, *H, W* and *k* are the height, width, and growth rate of the feature maps.

II. VGG-19

VGG-19 is a deep convolutional neural network with 19 layers. The network uses small 3×3 convolutional filters, which allow capturing fine spatial details while retaining computational efficiency. The architecture is characterized by alternating convolutional and max-pooling layers, followed by fully connected layers at the end.

Each convolutional layer applies a linear filter to the input feature map, followed by a non-linearity (typically ReLU). For a convolutional layer with filter weights *W* and biases *b*, the output at each layer is:

$$x_l = \sigma(W_l * x_{l-1} + b_l)$$

Here: * denotes the convolution operation, $\sigma(\cdot)$ is the activation function (ReLU), and x_{l-1} is the input feature map from the previous layer.

For a max-pooling layer with a pool size *p*, the output is:

$$xl = MaxPool(xl - 1, p)$$

This process is repeated for the 19 layers, where the final output from the convolutional stack is passed through fully connected layers to generate a feature vector.

III. InceptionNet-V3

InceptionNet-V3 introduces inception modules that allow the network to capture features at multiple scales. Each inception module applies multiple convolutions with different kernel sizes, pooling operations, and concatenates their outputs to create richer feature maps.

Let x be the input feature map to the inception module. The module computes several operations in parallel:

- 1×1 convolution: $x_{1\times1} = W_{1\times1} * x$
- 3×3 convolution: $x_{3\times3} = W_{3\times3} * x$
- 5×5 convolution: $x_{5\times5} = W_{5\times5} * x$
- Max-pooling operation: $x_{\text{pool}} = \text{MaxPool}(x, p)$

The outputs of these operations are concatenated to form the final output of the inception module:

$$x_{out} = [x_{1\times1}, x_{3\times3}, x_{3\times3}, x_{pool}]$$

InceptionNet-V3 stacks several such inception modules, followed by fully connected layers, to extract multi-scale features from the input.

Classification

Five different classification algorithms were employed to evaluate the effectiveness of the extracted features:

I. Support Vector Machine (SVM)

SVM is a supervised learning algorithm that seeks to find the optimal hyperplane that separates data points from different classes with the maximum margin. The algorithm can handle both linear and non-linear classification by using kernel tricks.

Given a dataset $\{(x_i, y_i)\}_{i=1}^{n}$, where *xi* represents the feature vectors and $y_i \in \{-1,1\}$ the labels, the goal of SVM is to find a hyperplane $w^T x + b = 0$ such that the margin between the classes is maximised:

$$\text{Minimise } \frac{1}{2}||w||^2 \text{ subject to } y_i(w^T x_i + b) \geq 1$$

For non-linear problems, the data can be transformed into higher dimensions using a kernel function $K(x_i, x_j)$.

II. Decision Tree Classifier (DTC)

A decision tree classifier is a tree-like model where each internal node represents a decision based on a feature, each branch represents the outcome of that decision, and each leaf node represents a class label. It partitions the feature space into regions and makes predictions by traversing the tree.

At each node, the algorithm selects the feature f_j that best splits the data based on an impurity measure such as Gini impurity or entropy:

$$\text{Gini impurity: } G(f_j) = 1 - \sum_{k=1}^{K} p_k^2$$

$$\text{Entropy: } H(f_j) = -\sum_{k=1}^{K} p_k \log(p_k)$$

Here, p_k represents the proportion of samples belonging to class *k*.

The algorithm recursively splits the data until a stopping criterion is met (*e.g.*, maximum depth, minimum samples per node, etc.).

III. Stochastic Gradient Descent (SGD) Classifier

The Stochastic Gradient Descent (SGD) classifier is a linear model that is trained using stochastic gradient descent, a variant of gradient descent where the model is updated for each training sample rather than for the entire dataset. This makes it highly efficient for large-scale data.

The objective is to minimise a loss function, such as the hinge loss for SVM or the logistic loss for logistic regression, using gradient descent:

$$w^{(t+1)} = w^{(t)} - \eta \nabla L\left(w^{(t)}, x_i, y_i\right)$$

Where, $w^{(t)}$ is the weight vector at iteration *t*, η is the learning rate, *L* is the loss function, and ∇L is the gradient of the loss function with respect to *w*.

In SGD, the model is updated for each sample (x_i, y_i), making it computationally efficient.

IV. Random forest classifier

Random forest is an ensemble learning method that combines multiple decision trees to make a more accurate and robust classifier. Each tree in the forest is trained on a random subset of the data, and the final prediction is made by averaging the predictions of all trees (for regression) or taking a majority vote (for classification). The prediction of a random forest is:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x)$$

Where, *T* is the number of decision trees, and $f_t(x)$ is the prediction from the *t*-th tree.

Each tree is built by recursively splitting the data, using an impurity measure like Gini impurity or entropy.

V. Logistic regression

Logistic regression is a linear model used for binary classification tasks. The model uses the logistic function to map the output to a probability between 0 and 1.

Given input features *x*, the logistic regression model predicts the probability $P(y = 1|x)$ using the logistic function:

$$P(y=1|x) = \frac{1}{1+e^{-(w^T x+b)}}$$

Where, w is the weight vector, x is the input feature vector, and b is the bias term.

The model is trained by minimising the log-likelihood loss function:

$$L(w,b) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log P(y_i|x_i) + (1-y_i)\log(1-P(y_i|x_i))]$$

Where, n is the number of training samples, and y_i is the true label for the i-th sample.

Each classification algorithm was trained and evaluated using the 'x features' extracted from each of the three transfer learning models. This allowed for a comprehensive comparison of the performance of different combinations of feature extraction and classification techniques.

Evaluation metrics

Let's define the following: TP: Number of correctly classified positive instances; TN: Number of correctly classified negative instances; FP: Number of negative instances incorrectly classified as positive; and FN: Number of positive instances incorrectly classified as negative.

Using these, we can express the core evaluation metrics mathematically:

Accuracy (ACC): Represents the overall proportion of correctly classified instances.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision (P): Quantifies the accuracy of positive predictions, indicating the proportion of true positives among all positive predictions.

$$P = \frac{TP}{TP+FP}$$

Accuracy and precision of all algorithms are depicted in Figure 3.

Recall (R): Measures the model's ability to identify all actual positive instances, representing the proportion of true positives among all actual positive instances.

$$R = \frac{TP}{TP+FN}$$

F1-Score (F1): Provides a harmonic mean between precision and recall, offering a balanced measure, especially valuable in scenarios with imbalanced class distributions.

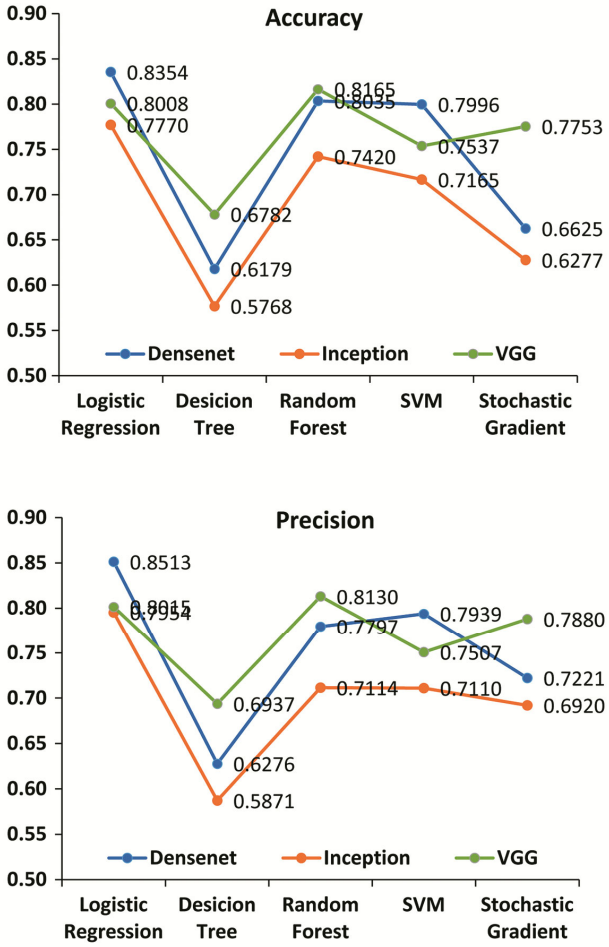

Fig. 3 — Accuracy and precision of all deep learning algorithms

$$F1 = 2 \cdot \frac{P \cdot R}{P+R}$$

F1-score and recall of all algorithms are depicted in Figure 4.

Materials and methods – Stepwise summary

1. Dataset:
   - 1,582 coral images from Kaggle (Great Barrier Reef).
   - Excluded dead corals → final dataset: 580 bleached, 570 unbleached images.
2. Pre-processing:
   - Images resized to 128×128 pixels.
   - Pixel values normalized using min-max normalisation.
3. Feature extraction (Transfer learning models):
   - DenseNet: Uses dense layer connections to enhance feature reuse.

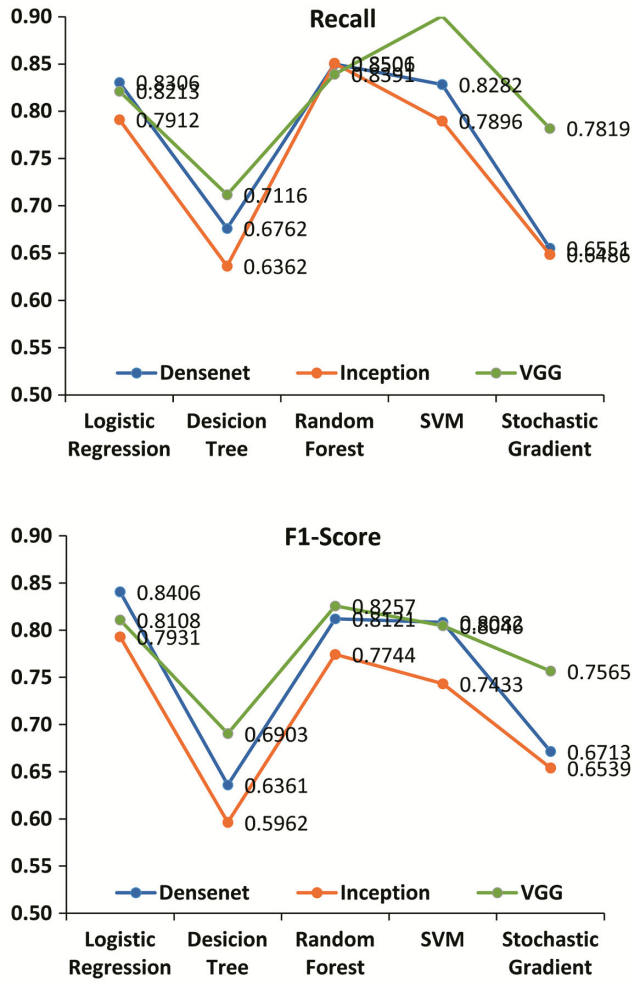

Fig. 4 — Recall and F1-score of all deep learning algorithms

- VGG-19: Deep CNN with 3×3 filters and ReLU activations.
- InceptionNet-V3: Uses inception modules for multi-scale feature capture.

4. Classification algorithms:
- SVM: Finds optimal hyperplane for class separation.
- Decision tree: Splits based on impurity (Gini/Entropy).
- SGD classifier: Linear model optimized with stochastic gradient descent.
- Random forest: Ensemble of decision trees for robust classification.
- Logistic regression: Predicts class probabilities using the logistic function.

5. Evaluation metrics:
- Accuracy, precision, recall, and F1-score, used for performance analysis.
- Results visualized through graphs (Figs. 3 & 4).

Results

Each of the five classifiers was paired with deep learning algorithms to compare their accuracy and outcomes, aiming to find the best combinations.

For logistic regression, solvers like Saga, Newton, Liblinear, and lbfgs were used with l1 and l2 penalties. The solver's Coordinate Descent algorithm addresses optimisation through sequential minimization along coordinate directions.

For the Support Vector Machine (SVM), different kernels (including polynomial) with varying degrees and gamma values were tested. The C parameter, ranging from 0 to 5, controlled the balance between misclassification prevention and margin size.

The Stochastic Gradient Descent (SGD) classifier utilized different penalties and loss functions, optimizing learning rate and loss parameters like Epsilon-insensitive, squared-hinge, and log for maximum accuracy.

Random forest and decision tree classifiers tested various tree splits, depths, and criteria such as entropy and Gini.

Finally, accuracy, F1-score, precision, and recall were calculated for every feature extraction and classification combination to evaluate performance.

Discussion

Coral bleaching is influenced by various factors such as the Bleaching Time Index (BTI), Sea Surface Temperature (SST), marine heatwaves, prolonged positive temperature anomalies, and total organic carbon levels. Regardless of these contributing factors, method in the current study focuses on classifying coral images into two categories: bleached and unbleached.

The dataset in this study comprises of a balanced set of images with 50.4 % of bleached images and 49.6 % of non-bleached images.

The objective of the proposed method is to explore and evaluate five distinct classification algorithms – Logistic regression, decision tree, random forest, SVM, and stochastic gradient descent, each paired with three prominent feature extractors: DenseNet, Inception, and VGG. Model performance was assessed by averaging the accuracy achieved over multiple training epochs.

Here, Table 2 illustrates the performance of a Support Vector Machine (SVM) classifier evaluated on three distinct models *i.e.* inception, VGG, and DenseNet across a range of hyper-parameter settings.

Table 2 — Evaluation of the accuracy, F1-score, recall and precision of various deep learning algorithms in combination with Support Vector Machine (SVM) algorithm, considering all parameter configurations

Algorithm	C	Kernel	Degree	Gamma	Accuracy	F1-score	Recall	Precision
Inception	0.1	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.6887	0.7272	0.8000	0.6917
Inception	0.1	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.7131	0.7467	0.8062	0.7067
Inception	0.5	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.7364	0.7588	0.8003	0.7239
Inception	1	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.7298	0.7527	0.7845	0.7251
Inception	5	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.7146	0.7309	0.7571	0.7074
VGG	0.1	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.6524	0.7638	0.9816	0.6481
VGG	0.1	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.6830	0.7719	0.9439	0.6782
VGG	0.5	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.7704	0.7998	0.8525	0.7657
VGG	1	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.7952	0.8150	0.8470	0.7943
VGG	5	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.8673	0.8726	0.8783	0.8671
DenseNet	0.1	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.7580	0.7649	0.7655	0.7739
DenseNet	0.1	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.8028	0.8142	0.8360	0.7969
DenseNet	0.5	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.8225	0.8364	0.8765	0.8035
DenseNet	1	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.8156	0.8269	0.8554	0.8027
DenseNet	5	rbf, poly, linear, sigmoid	2, 3, 4	auto, scale	0.7991	0.7986	0.8077	0.7923

Table 3 — Evaluation of the accuracy, F1-score, recall and precision of various deep learning algorithms in combination with Decision Tree (DT) algorithm, considering all parameter configurations

Algorithm	Criterion	Splitter	max_depth	max_features	Accuracy	F1-score	Recall	Precision
Inception	gini	Best	2,3,4,5	sqrt, log2	0.5753	0.6136	0.6866	0.5681
Inception	gini	random	2,3,4,5	sqrt, log2	0.5793	0.5819	0.6054	0.5876
Inception	gini	best	2,3,4,5	sqrt, log2	0.5873	0.6213	0.7000	0.5758
Inception	entropy	random	2,3,4,5	sqrt, log2	0.5669	0.5860	0.6170	0.5707
Inception	log_loss	Best	2,3,4,5	sqrt, log2	0.6014	0.6308	0.6750	0.6304
Inception	log_loss	random	2,3,4,5	sqrt, log2	0.5505	0.5438	0.5330	0.5901
VGG	gini	best	2,3,4,5	sqrt, log2	0.7263	0.7345	0.7438	0.7351
VGG	gini	random		sqrt, log2	0.6375	0.6369	0.6497	0.6725
VGG	entropy	best	2,3,4,5	sqrt, log2	0.7003	0.6978	0.6858	0.7225
VGG	entropy	random	2,3,4,5	sqrt, log2	0.6446	0.6782	0.7287	0.6516
VGG	log_loss	best	2,3,4,5	sqrt, log2	0.7081	0.7427	0.8066	0.6954
VGG	log_loss	random	2,3,4,5	sqrt, log2	0.6523	0.6517	0.6548	0.6851
DenseNet	gini	best	2,3,4,5	sqrt, log2	0.6226	0.6477	0.6923	0.6249
DenseNet	gini	random	2,3,4,5	sqrt, log2	0.6042	0.5940	0.5827	0.6678
DenseNet	entropy	best	2,3,4,5	sqrt, log2	0.6290	0.6358	0.6643	0.6397
DenseNet	entropy	random	2,3,4,5	sqrt, log2	0.6015	0.6294	0.6851	0.5922
DenseNet	log_loss	best	2,3,4,5	sqrt, log2	0.6368	0.6641	0.7161	0.6271
DenseNet	log_loss	best	2,3,4,5	sqrt, log2	0.6135	0.6455	0.7167	0.6142

Based on this data, VGG with C = 5 seems to provide the best overall performance across all metrics. Dense Net shows competitive results, particularly at C = 0.5, while inception delivers solid but slightly lower performance. The choice of kernel (rbf, poly, linear, sigmoid) and the value of C significantly affect the SVM's classification performance, with higher C values generally favoring better model performance up to a certain point.

Unlike SVM, the evaluation of decision tree classifier, stochastic gradient, random forest and logistic regression is also presented in Tables 3, 4, 5 and 6, respectively. In decision tree classifier, VGG consistently outperforms the other models, particularly with gini and log_loss criteria. Inception and densenet show moderate performance, with log_loss yielding the best results for both. In Table 4, densenet shows moderate performance, peaking at alpha = 0.5 with accuracy 0.6820. But in Table 6, densenet consistently achieves the highest performance across metrics, peaking with C = 1 (accuracy 0.8385, F1-score 0.8435), maintaining strong precision and recall. DenseNet outperforms VGG and inception in logistic regression classifier analysis, especially with larger C values. VGG delivers competitive results, while inception shows moderate performance across all metrics. Table 7 illustrates the overall average accuracy across all deep learning models and classification algorithms.

Table 4 — Evaluation of accuracy, F1-score, recall and precision of various deep learning algorithms in combination with Stochastic Gradient Descent (SGD) algorithm with all parameter configurations

Algorithm	Alpha	Learning rate	Loss	Accuracy	F1-score bleached	Recall bleached	Precision bleached
DenseNet	0.001	adaptive, invscaling, constant	squared_ epsilon_ insensitive, huber, epsilon_ insensitive, squared_ epsilon_ insensitive hinge log_loss modified_ huber squared_ hinge perception squared_ error	0.6506	0.6621	0.6393	0.7260
DenseNet	0.0001	adaptive, invscaling, constant		0.6690	0.6852	0.6752	0.7144
DenseNet	0.005	adaptive, invscaling, constant		0.6659	0.6701	0.6482	0.7203
DenseNet	0.00001	adaptive, invscaling, constant		0.6585	0.6751	0.6564	0.7164
DenseNet	0.0005	adaptive, invscaling, constant		0.6587	0.6856	0.6673	0.7294
DenseNet	0.1	adaptive, invscaling, constant		0.6639	0.6566	0.6298	0.7342
DenseNet	0.5	adaptive, invscaling, constant		0.6820	0.7013	0.7007	0.7321
DenseNet	1	adaptive, invscaling, constant		0.6641	0.6453	0.6386	0.7203
DenseNet	0.00005	adaptive, invscaling, constant		0.6499	0.6604	0.6400	0.7058
Inception	0.001	adaptive, invscaling, constant		0.5995	0.6242	0.6016	0.6712
Inception	0.0001	adaptive, invscaling, constant		0.6215	0.6617	0.6603	0.6848
Inception	0.005	adaptive, invscaling, constant		0.6554	0.6853	0.6798	0.7083
Inception	0.00001	adaptive, invscaling, constant		0.6315	0.6620	0.6534	0.6937
Inception	0.0005	adaptive, invscaling, constant		0.6287	0.6399	0.6260	0.6927
Inception	0.1	adaptive, invscaling, constant		0.6232	0.6514	0.6365	0.7148
Inception	0.5	adaptive, invscaling, constant		0.6365	0.6407	0.6480	0.7006
Inception	1	adaptive, invscaling, constant		0.6230	0.6644	0.6885	0.6740
Inception	0.00005	adaptive, invscaling, constant		0.6296	0.6555	0.6433	0.6878
VGG	0.00001	adaptive, invscaling, constant		0.9877	0.9881	0.9877	0.9885
VGG	0.00005	adaptive, invscaling, constant		0.9771	0.9780	0.9831	0.9733
VGG	0.0001	adaptive, invscaling, constant		0.9507	0.9522	0.9525	0.9540
VGG	0.0005	adaptive, invscaling, constant		0.9036	0.9074	0.9156	0.9039
VGG	0.001	adaptive, invscaling, constant		0.8458	0.8500	0.8514	0.8632
VGG	0.005	adaptive, invscaling, constant		0.7539	0.7476	0.7341	0.8149
VGG	0.1	adaptive, invscaling, constant		0.5729	0.6051	0.7200	0.6276
VGG	0.5	adaptive, invscaling, constant		0.5066	0.5555	0.6751	0.5367
VGG	1	adaptive, invscaling, constant		0.4796	0.2249	0.2176	0.4299

Comparatively, other model combinations demonstrated varying degrees of success. The random forest classifier, when paired with densenet and VGG, achieved respectable accuracies of 80 % and 81 %, respectively. This highlights the robustness of ensemble methods like random forest in handling complex datasets and their ability to generalise well. However, the relatively lower performance of decision tree and stochastic gradient descent across all feature extractors underscores the potential limitations of these algorithms when dealing with intricate image data and the need for careful hyperparameter tuning.

As part of this growing research, various machine learning approaches have been explored to enhance coral classification. In this context, the current study and Jamil *et al*.[24] utilised the BHD dataset for coral bleaching detection, focusing on classifying images as bleached and unbleached. Current study used 1150 images (580 bleached and 570 unbleached) and employed a dense net-logistic regression model, achieving 83 % accuracy, an F1-score of 0.84, and a precision of 0.85. Jamil *et al*.[24] used a Bag of Features (BoF) model combined with deep learning and reported higher accuracy (above 90 %). While the BHD dataset contains 1582 images divided in three categories *viz.* bleached, dead, and unbleached, their study did not explicitly use the three-class approach, instead focusing on binary classification alike current study. The BoF model likely benefited from capturing localised image features, whereas our dense net-logistic regression model emphasised computational efficiency and effective feature extraction. Both studies relied solely on image-based data without integrating environmental factors (*e.g.*, BTI, SST, marine heatwaves), emphasising a shared limitation. Despite the differences, the model used in the current study demonstrated competitive performance, reinforcing its potential for streamlined and accurate coral bleaching assessments.

Table 5 — Evaluation of the accuracy, F1-score, recall and precision of various deep learning algorithms in combination with Random Forest (RT) algorithm, considering all parameter configurations

Algorithm	Criterion	max_ depth	n estimators	max_ features	Accuracy	F1-score	Recall	Precision
Inception	gini	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	sqrt	0.7555	0.7887	0.8791	0.7154
Inception	gini	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	log2	0.7285	0.7612	0.8296	0.7039
Inception	entropy	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	sqrt	0.7518	0.7848	0.8750	0.7118
Inception	entropy	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	log2	0.7350	0.7657	0.8237	0.7161
Inception	log_loss	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	sqrt	0.7506	0.7831	0.8714	0.7114
Inception	log_loss	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	log2	0.7308	0.7629	0.8247	0.7101
VGG	gini	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	sqrt	0.8295	0.8372	0.8468	0.8279
VGG	entropy	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	sqrt	0.8276	0.8351	0.8438	0.8266
VGG	log_loss	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	sqrt	0.8270	0.8345	0.8430	0.8262
VGG	gini	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	log2	0.8071	0.8176	0.8343	0.8018
VGG	entropy	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	log2	0.8040	0.8151	0.8336	0.7977
VGG	log_loss	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	log2	0.8039	0.8149	0.8331	0.7977
DenseNet	gini	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	log2	0.7850	0.7879	0.7916	0.7852
DenseNet	entropy	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	log2	0.7550	0.7445	0.7357	0.7536
DenseNet	gini	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	sqrt	0.8164	0.8273	0.8776	0.7828
DenseNet	entropy	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	sqrt	0.7853	0.7856	0.7908	0.7806
DenseNet	log_loss	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	sqrt	0.8315	0.8490	0.9219	0.7868
DenseNet	log_loss	2, 3, 4, 5	50, 100, 150, 200, 250, 400, 500	log2	0.7759	0.7960	0.8787	0.7278

Table 6 — Evaluation of the accuracy, F1-score, recall and precision of various deep learning algorithms in combination with Logistic Regression (LR) algorithm, considering all parameter configurations

Algorithm	C	Solver	Penalty	Accuracy	F1-score	Recall	Precision
Inception	0.05	liblinear, saga, newton-cg, lbfgs	l1, l2	0.7788	0.7983	0.8061	0.7914
Inception	0.1	liblinear, saga, newton-cg, lbfgs	l1, l2	0.7761	0.7953	0.7990	0.7921
Inception	0.5	liblinear, saga, newton-cg, lbfgs	l1, l2	0.7743	0.7900	0.7867	0.7934
Inception	1	liblinear, saga, newton-cg, lbfgs	l1, l2	0.7761	0.7894	0.7786	0.8007
Inception	5	liblinear, saga, newton-cg, lbfgs	l1, l2	0.7798	0.7926	0.7857	0.7997
VGG	0.05	liblinear, saga, newton-cg, lbfgs	l1, l2	0.8036	0.8154	0.8390	0.7946
VGG	0.1	liblinear, saga, newton-cg, lbfgs	l1, l2	0.8057	0.8172	0.8409	0.7964
VGG	0.5	liblinear, saga, newton-cg, lbfgs	l1, l2	0.7992	0.8080	0.8149	0.8015
VGG	1	liblinear, saga, newton-cg, lbfgs	l1, l2	0.7987	0.8064	0.8071	0.8057
VGG	5	liblinear, saga, newton-cg, lbfgs	l1, l2	0.7969	0.8069	0.8045	0.8093
DenseNet	0.01	liblinear, saga, newton-cg, lbfgs	l1, l2	0.8272	0.8358	0.8321	0.8407
DenseNet	0.05	liblinear, saga, newton-cg, lbfgs	l1, l2	0.8349	0.8418	0.8357	0.8488
DenseNet	1	liblinear, saga, newton-cg, lbfgs	l1, l2	0.8385	0.8435	0.8314	0.8560
DenseNet	1.5	liblinear, saga, newton-cg, lbfgs	l1, l2	0.8369	0.8415	0.8286	0.8549
DenseNet	5	liblinear, saga, newton-cg, lbfgs	l1, l2	0.8365	0.8399	0.8257	0.8546
DenseNet	10	liblinear, saga, newton-cg, lbfgs	l1, l2	0.8381	0.8411	0.8300	0.8526

Future directions

While the results of present study demonstrate the potential of deep learning for automated coral bleaching identification, further investigation is warranted.

Future research could explore:

- Expanding the dataset:

A larger and more diverse dataset consisting of various coral species, bleaching stages and underwater environments would enhance model robustness and generalisability.

- Ensemble techniques:

Investigating the efficacy of combining multiple high-performing models through ensemble techniques could potentially lead to even more accurate and reliable bleaching detection.

Table 7 — Average accuracy of all deep learning algorithms with combination of classification algorithms

	Average	Accuracy	F1-score	Recall	Precision
VGG	Logistic regression	0.8008	0.8108	0.8213	0.8015
	Decision tree	0.6782	0.6903	0.7116	0.6937
	Random forest	0.8165	0.8257	0.8391	0.8130
	SVM	0.7537	0.8046	0.9007	0.7507
	Stochastic gradient	0.7753	0.7565	0.7819	0.7880
Inception	Logistic regression	0.7770	0.7931	0.7912	0.7954
	Decision tree	0.5768	0.5962	0.6362	0.5871
	Random forest	0.7420	0.7744	0.8506	0.7114
	SVM	0.7165	0.7433	0.7896	0.7110
	Stochastic gradient	0.6277	0.6539	0.6486	0.6920
DenseNet	Logistic regression	0.8354	0.8406	0.8306	0.8513
	Decision tree	0.6179	0.6361	0.6762	0.6276
	Random forest	0.8035	0.8121	0.8501	0.7797
	SVM	0.7996	0.8082	0.8282	0.7939
	Stochastic gradient	0.6625	0.6713	0.6551	0.7221

Conclusion

This research contributes to the growing body of knowledge leveraging deep learning for coral reef monitoring and conservation. By automating the identification of bleached corals, timely interventions can be facilitated and can contribute to the preservation of these vital ecosystems. This study investigated the effectiveness of various deep learning models for the automated identification of bleached coral images. Utilising a publicly available balanced dataset of 1150 images, this study evaluated the performance of five classification algorithms in conjunction with three feature extractors.

The findings demonstrate superior performance of the densenet-logistic regression model, achieving the highest average accuracy (83 %), F1-score (0.84), and precision (0.85). This suggests that the densenet architecture effectively captures intricate features indicative of coral bleaching, while logistic regression provides robust classification capabilities.

While this research highlights the potential of deep learning for automating coral bleaching detection, further investigation with larger, more diverse datasets is crucial for developing highly accurate and generalisable models. Developing accurate automated tools for identifying bleached corals is vital for large-scale monitoring and conservation. This study adds to the growing research in this area, providing a foundation for future work using deep learning to protect coral reef ecosystems.

Conflict of Interest

Authors declare no competing or conflict of interest.

Author Contributions

JB: Streamlined the entire process of experimentation and documentation of results & literature survey; MT: Initialised the task of experimentation through various IP techniques; and GS: Located a strong dataset, tabulation of the results through various iterations.

References

1 Wikipedia, *Coral reef*, World Wide Web electronic publication, Accessed Online at: https://en.wikipedia.org/w/index.php?title=Coral_reef&oldid=1216440795; (Accessed on 09/2024).

2 Coral Reef Alliance, *Coral Reefs Ecology & Biodiversity*, World Wide Web electronic publication, Accessed Online at: https://coral.org/en/coral-reefs-101/; (Accessed on 09/2024).

3 NOAA, *What is coral bleaching?* World Wide Web electronic publication, Accessed Online at: https://oceanservice.noaa.gov/facts/coral_bleach.html; (Accessed on 08/2024).

4 Spady B L, Skirving W J, Liu G, De La Cour J L, McDonald C L, *et al.*, Unprecedented early-summer heat stress and forecast of coral bleaching on the Great Barrier Reef, 2021-2022, *F1000 Res*, 11 (2022) p. 127. https://doi.org/10.12688/f1000research.108724.4

5 NOAA, *NOAA Coral Reef Watch 5 km Methodology*, World Wide Web electronic publication, Accessed Online at: https://coralreefwatch.noaa.gov/product/5km/methodology.php; (Accessed on 09/2024)

6 McGowan H & Theobald A, A typical weather patterns cause coral bleaching on the Great Barrier Reef, Australia during the 2021–2022 La Niña, *Sci Rep*, 13 (2023) p. 1-8. https://doi.org/10.1038/s41598-023-33613-1

7 Lough J M, Anderson K D & Hughes T P, Increasing thermal stress for tropical coral reefs: 1871–2017, *Sci Rep*, 8 (2018) 1-8 (Art No 6079). https://doi.org/10.1038/s41598-018-24530-9

8 Australian Institute of Marine Science, *Coral bleaching events*, *AIMS*, 2024, Available Online at: https://www.aims.gov.au/

research-topics/environmental-issues/coral-bleaching/coral-bleaching-events; (Accessed on 08/2024)

9 Hoegh-Guldberg O, Jacob D, Taylor M, Bindi M, Brown S, *et al*., Impacts of 1.5°C global warming on natural and human systems, In: *Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty*, edited by Masson-Delmotte V, Zhai P, Pörtner H-O, Roberts D, Skea J, *et al*., (Cambridge University Press, Cambridge, UK and New York, NY, USA), pp. 175-312. https://doi.org/10.1017/9781009157940.005

10 Wang J-T, Chu C-W & Soong K, Comparison of the bleaching susceptibility of coral species by using minimal samples of live corals, *PeerJ*, 10 (2022) p. e12840. https://doi.org/10.7717/peerj.12840

11 Putra R D, Suhana M P, Kurniawn D, Abrar M, Siringoringo R M, *et al*., Detection of reef scale thermal stress with Aqua and Terra MODIS satellite for coral bleaching phenomena, *Proc AIP Conf*, 2094 (1) (2019) p. 01. https://doi.org/10.1063/1.5097493

12 Genevier L G C, Jamil T, Raitsos D E, Krokos G & Hoteit I, Marine heatwaves reveal coral reef zones susceptible to bleaching in the Red Sea, *Global Change Biol*, 25 (7) (2019) 2338–2351. https://doi.org/10.1111/gcb.14652

13 Mohit A, Nandini Ray C, Ashwin G, Harshad P, Rakesh P, *et al.*, Coral bleaching due to increased sea surface temperature in Gulf of Kachchh Region, India, during June 2016, *Indian J Geo-Mar Sci*, 48 (3) (2019) 327–332.

14 Kalita S, Kumar S, Sarma H P & Devi A, Total organic carbon, heavy metal content and metal bioaccumulation in a freshwater wetland of Indo-Burmese province, India, *Int J Environ Anal Chem*, 103 (16) (2023) 4488–4502. https://doi.org/10.1080/03067319.2021.1928104

15 Torres-Aguirre E, Flores-de-Santiago F, Valderrama-Landeros L, Amezcuac F & Flores-Verdugo F, Semiarid mangrove species classification using machine learning algorithms and visible UAV data, *Indian J Geo-Mar Sci*, 53 (03) (2024) 109-118. https://doi.org/10.56042/ijms.v53i03.8184

16 Jayakumar K, Spatio-temporal changes in the Godavari mangroves: A study examining land-use change and sustainable management, *Indian J Geo-Mar Sci*, 53 (01) (2023) 15-23. https://doi.org/10.56042/ijms.v52i01.5436

17 Dey J & Vijay R, Geospatial assessment of seasonal water quality of Malad creek, Mumbai, India: An impact of sewage discharge, *Indian J Geo-Mar Sci*, 51 (10) (2022) 811-821. https://doi.org/10.56042/ijms.v51i10.2930

18 Bharath M S, Chandran R, Senthilkumaran R, Ramkumarana K, Thanappana V P P, *et al*., Preliminary report on the occurrence of lethal crustose (non-geniculate) coralline algae white band and white patch diseases in the coral reefs of Gulf of Kachchh, India, *Indian J Geo-Mar Sci*, 52 (05) (2023) 255-260. https://doi.org/10.56042/ijms.v52i05.8010

19 Manoj R V, Rajani P R, Girish Kumar M, Gopakumar B, Joshie R K, *et al.*, Application of liberation analysis in ferromanganese encrustations from South Andaman Sea: An automated mineral identification technique using TIMA, *Indian J Geo-Mar Sci*, 51 (11) (2022) 918-929. https://doi.org/10.56042/ijms.v51i11.3509

20 Das K, Janardhana P & Narayana H, Application of CNN based image classification technique for oil spill detection, *Indian J Geo-Mar Sci*, 52 (01) (2023) 05-14. https://doi.org/10.56042/ijms.v52i01.5438

21 Rodríguez-Sobreyra R, Álvarez-Sánchez L F & Flores-de-Santiago F, Feasibility of an open-source algorithm for predicting sea surface temperature based on three multi-resolution data sources, *Indian J Geo-Mar Sci*, 52 (06) (2023) 284-291. https://doi.org/10.56042/ijms.v52i06.8349

22 Kaggle, *BHD Corals*, World Wide Web electronic publication, Accessed Online at: https://www.kaggle.com/datasets/sonainjamil/bhd-corals; (Accessed on 12/2023).

23 Jamil S, Rahman M & Haider A, Bag of Features (BoF) Based Deep Learning Framework for Bleached Corals Detection, *Big Data Cogn Comput*, 5 (4) (2021) 1-15 (Art No 53). https://doi.org/10.3390/bdcc5040053