Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

A Comparative Evaluation of Convolutional and Transformer Architectures for Underwater Coral Reef Classification

John Bennet Johnson[1]* and Radhakrishnan Vignesh[2]

[1,2]Presidency School of Artificial Intelligence & Advanced Computing, Presidency University, Bengaluru 560119, Karnataka, India.
Email: john.bennet@presidencyuniversity.in ; john.20223CSE0022@presidencyuniversity.in; vigneshr@presidencyuniversity.in
*Corresponding Author: John Bennet Johnson

Abstract
Coral reefs are among the most biodiverse and most rapidly declining marine ecosystems, and automated classification of reef imagery underpins biodiversity assessment, bleaching surveillance and conservation planning. Seven widely used transfer-learning backbones EfficientNetB0, ResNet50, ResNet101, ResNet152V2, VGG16, VGG19 and Xception are evaluated under a fixed train, validation and test partition, and four modern architectures ResNet50, EfficientNet-B3, ViT-Base and DenseNet121 under stratified 13-fold cross-validation with out-of-fold aggregation; ResNet50 appears in both, allowing the sensitivity of a single architecture to the evaluation protocol to be observed directly. Overall accuracy is a weak guide to practical utility every architecture recovers the morphologically distinctive classes almost perfectly while failing on the same three visually ambiguous classes. Second, those failures are only partially correlated across architectures a class recovered at 0.84 recall by one model is recovered at 0.13 by another indicating substantial headroom for ensemble methods and identifying precisely which classes future work should target.

Keywords: Coral reef classification, Comparative evaluation, Benchmarking, Vision Transformer, Convolutional neural network, Underwater image analysis.

1 Introduction

High-quality underwater visual data are needed to quantify biodiversity loss, coral bleaching and climate-driven habitat degradation, all of which are now matters of global concern. Wavelength-dependent absorption and scattering produce haze, low contrast and pronounced colour distortion, which together make reef imagery difficult to interpret both visually and computationally [1].

Underwater image degradation occurs because two physical processes absorption and scattering respectively cause light to behave in different ways. Longer wavelengths like red and orange get absorbed through water according to their specific absorption rates but shorter wavelengths like blue and green light can travel through water at greater distances. Underwater images show a bluish and greenish colour because they lose their warm colour elements [1]. Scattering happens when light interacts with particles that float in the water like sand and plankton and organic material. Forward scattering causes blurring and loss of detail, while backscattering introduces veiling light that drastically reduces contrast [18].

The testing results show that coral classification algorithms which use terrestrial-quality images fail to perform correctly when they process underwater distortion. The poor-quality images produce negative effects on object detection tasks and seafloor mapping processes and biological species identification activities according to [5]. Underwater image enhancement establishes itself as an essential preprocessing method which enhances the accuracy and reliability of future research evaluations.

Researchers have investigated traditional image enhancement techniques which include histogram equalization and Retinex-based filtering and white-balancing methods for underwater restoration purposes. The Retinex-based models use reflectance separation from illumination to create human vision simulation whereas histogram equalization achieves better visual contrast through pixel intensity distribution [13]. The methods demonstrate incapacity to operate correctly because they suffer from excessive enhancement and shift colours and cannot function properly in multiple underwater environments [14]. Linear white-balancing methods often assume uniform illumination, an assumption almost never true in underwater environments.

Deep learning has brought major changes to the field of underwater image enhancement. Data-driven models have shown exceptional ability to understand complex nonlinear relationships which exist between degraded images and their respective enhanced versions. Researchers have widely applied convolutional neural networks (CNNs) together with encoder–decoder frameworks such as U-Net and ResUNet to restore underwater images by using paired datasets that contain both degraded images and their corresponding ground-truth images [3].

Researchers have increasingly adopted unpaired learning models because they need more paired data. Underwater colour correction and contrast enhancement processes use CycleGAN because it operates without the need for paired data according to results from [6] and [12]. GAN models learn high-quality underwater image style distribution while transforming low-quality images into their cleaner versions. WaterGAN synthesizes realistic underwater images from depth maps which allows the model to learn enhancement without needing actual paired data according to [7]. GAN-based systems create visually appealing results which maintain perceptual realism; however, they produce artifacts and fail to maintain structural details.

The recent survey research and benchmark studies demonstrate the requirement for an integrated approach that merges the advantages of both supervised and unsupervised deep learning techniques [5], [11] and [20]. Two observations follow from this body of work. First, underwater enhancement research is mature but is almost always evaluated on image-quality proxies such as PSNR, SSIM, UIQM and UCIQE rather than on the recognition task it is ultimately meant to serve. Second, where recognition is addressed directly, results are difficult to compare across studies: authors report on different datasets, with different numbers of classes, under different partitioning schemes and with different averaging conventions for precision and recall. A practitioner choosing an architecture for reef monitoring therefore has little reliable basis for the decision.

Holding the data fixed also makes a second question tractable. Aggregate accuracy is the metric almost universally reported, but it conceals which classes a model actually fails on. In a conservation setting the distinction matters: a model that is 85% accurate overall but cannot separate bleached from dead coral is of limited operational value, because that specific boundary is the one bleaching surveillance depends on. This study therefore reports per-class behaviour throughout and treats the error structure, not the headline number, as the primary object of analysis.

Ten distinct architectures are evaluated across eleven evaluations on the same eight-class coral dataset. Seven established transfer-learning backbones are assessed under a fixed partition, and four modern convolutional and transformer architectures under stratified 13-fold cross-validation with out-of-fold aggregation. ResNet50 is common to both groups, so its two results also indicate how sensitive a single architecture is to the evaluation protocol. Because the two groups were trained under different partitioning schemes, they are reported separately within Section 4 and are not merged into a single ranking; comparisons are drawn within each scheme, and the cross-protocol discussion is restricted to per-class error patterns, which are protocol-independent.

The principal contributions of this work are as follows.

- A controlled comparative evaluation of ten convolutional and transformer architectures on a single curated eight-class coral dataset, holding data, class definitions and evaluation metrics constant so that architectural differences are isolated from experimental ones.
- A per-class analysis showing that aggregate accuracy conceals a consistent failure mode: every architecture recovers Acropora, foliose and staghorn corals almost perfectly while degrading sharply on branching soft, bleached and lobe corals, which are the classes of greatest ecological interest.
- Evidence that these failures are only partially correlated across architectures, establishing that the residual errors are architecture-specific rather than intrinsic to the data, and quantifying the headroom available to ensemble and multi-model approaches.

2 Related Work

Underwater imaging is difficult because the aquatic medium has optical properties fundamentally different from air. The dominant degradation factors are wavelength-dependent absorption, forward and backward scattering, non-uniform illumination and noise from suspended particulates. Interactions between incident light, water molecules and dissolved organic matter alter light propagation and compromise both the spatial and the spectral fidelity of the captured image. These mechanisms frame the literature reviewed below, since every family of methods classical, physics-based, CNN, GAN and hybrid targets a specific subset of them.

The process of absorption causes longer wavelengths to be selectively diminished through red and orange and yellow light as depth increases which creates a blue and green colour scheme that dominates the visual output. The backscattering process together with other scattering types creates a haze which functions as

a veil to decrease scene contrast while it hides the edges of objects. The process of forward scattering leads to image blurring which results in both decreased spatial resolution and missing structural information. Underwater lighting systems create spatially uneven illumination patterns because of both artificial lighting sources like strobes and flashlights and natural refraction caused by surface waves which produce complex luminance patterns. The development of advanced algorithms throughout the years has emerged because enhancement models need to handle three different types of distortions which include nonlinear responses and depth-related changes and pattern-based variability.

2.1 Classical Image Processing Approaches

Early underwater enhancement work adapted algorithms developed for low-light and low-contrast terrestrial imagery: histogram equalisation (HE), contrast-limited adaptive histogram equalisation (CLAHE), gamma correction, white balance and Retinex-based illumination correction. Fu et al. applied Retinex theory to underwater imaging, adjusting the illumination component while attempting to preserve scene reflectance [13]. Such algorithms are computationally inexpensive and simple to implement, but their operating envelope is narrow.

The system shows an inability to process changes because it maintains fixed light conditions and continuous material wear which do not reflect actual underwater conditions. HE-based methods create extra brightness through their processing methods which results in two types of defects: increased background noise and excessive colour brightness. The Retinex models show performance problems because they depend on specific lighting conditions which actual work environments do not provide. The models fail to include any physics-based elements because they do not recognize how underwater environments lose light through two processes: attenuation and scattering. The traditional methods helped scientists make initial discoveries, but they lacked the strength and broad applicability needed to solve contemporary underwater problems.

2.2 CNN-Based Approaches

Deep learning transformed underwater image analysis by making the degradation-to-restoration mapping learnable end to end. Most enhancement systems are built on convolutional neural networks (CNNs), typically arranged in encoder–decoder configurations.

UWCNN, an early supervised CNN for underwater enhancement [8]. UWCNN uses dense feature-extraction blocks to capture low- and mid-level cues that drive structural and chromatic fidelity, and is trained on a mixture of synthetic and real underwater data using weighted loss functions.

Anwar et al. created a residual-learning framework which uses hierarchical features to help the network learn both local and global enhancement information. The model achieved higher performance results on multiple underwater benchmark datasets because it successfully restored contrast while reducing bluish colour casts.

Supervised models face their main limitation because they need paired underwater datasets which do not exist. It is impossible to capture the same scene using both degraded conditions and ground-truth conditions. To solve this problem Li et al. developed WaterGAN which creates authentic underwater images through the use of depth maps extracted from land-based photographs [7]. WaterGAN enabled extensive supervised learning through its combination of physics-based modeling and data-driven learning methods.

2.3 GAN-Based Approaches

Generative Adversarial Networks (GANs) achieved a major leap in underwater image enhancement by enabling unpaired learning, texture synthesis, and domain translation.

CycleGAN established a framework for mapping between unpaired image domains under a cycle-consistency constraint. Arredondo et al. applied CycleGAN to underwater footage, learning a transformation from degraded to visually plausible underwater scenes without any matched image pairs [12].

Fabbri et al. proposed an underwater enhancement GAN that balances content preservation against perceptual improvement through adversarial training [6]. GAN-based methods restore natural colour tones and visual appeal effectively, but can hallucinate texture and distort structural detail.

Zhou et al. designed a multi-scale dense GAN that fuses features across resolution levels to improve both global contrast and fine detail [16]. Hou et al. combined GAN-based illumination correction with denoising to address low-light, high-noise underwater conditions [15].

2.4 Hybrid Deep Learning Approaches

Hybrid approaches combine the complementary strengths of CNNs, GANs, physics-based priors and attention mechanisms.

Liu et al. proposed a multi-branch architecture that couples perceptual loss with structure-preserving constraints and a color-restoration module, improving generalization to real-world underwater conditions [20]. Attention mechanisms allow the network to weight the most informative spatial regions, which improves the recovery of coral texture, edges and microstructure. Jiang et al. showed that ensemble learning can address the heterogeneity of underwater environmental conditions [11]. Combining several models through a meta-learner improves consistency and overall generalization.

2.5 Research Gap

Three observations follow from the literature reviewed above. First, the overwhelming majority of underwater deep learning work optimises image quality metrics such as PSNR, SSIM, UIQM and UCIQE, and treats downstream recognition accuracy as an assumed rather than a measured consequence. Second, where classification is addressed, it is almost always with a single backbone; comparative studies report which architecture wins on a given dataset but do not exploit the fact that different architectures fail on different images. Third, the ensemble work that does exist relies predominantly on homogeneous model families or on fixed combination rules such as majority voting and simple probability averaging, neither of which can learn that a particular base learner is systematically more reliable for a particular class. The present work addresses this gap by combining architecturally heterogeneous convolutional and transformer base learners under a trained probabilistic meta-learner, and by evaluating the result directly on species-level classification accuracy rather than on image-quality proxies.

3 Materials and Methods

3.1 Study Design

This work presents an end-to-end deep learning framework for underwater coral reef classification under degraded imaging conditions. Underwater acquisition introduces light absorption, forward and backward scattering, colour loss from suspended particles and non-uniform illumination; these defects reduce visibility, destroy chromatic information and obscure textural structure, and consequently depress the accuracy of automated classifiers. The proposed pipeline addresses them in sequence: image enhancement, preprocessing, feature learning, deep classification and evaluation. The workflow proceeds from dataset acquisition through to interpretation and comparative validation, and its three processing modules jointly improve the visual quality and the semantic separability of coral imagery.

The overall workflow consists of seven major stages: dataset acquisition, image enhancement, preprocessing and augmentation, deep feature extraction, convolutional neural network-based classification, training and optimization, and performance evaluation. The enhanced and normalized images are fed into a deep neural network that automatically learns hierarchical features representing coral morphology, texture, and colour characteristics. The model predictions are validated using statistical performance metrics and visual diagnostic tools. The step-by-step organization of this process establishes three essential qualities which enable scientists to repeat experiments and build systems that work in actual underwater monitoring environments.

3.2 Dataset Description and Acquisition

Experiments use the Enhancing Underwater Visual Perception (EUVP) dataset, a standard benchmark for underwater image enhancement and restoration. EUVP contains paired and unpaired underwater imagery captured under varied conditions including low light, turbidity, colour cast and motion blur. The paired subset provides degraded images together with high-quality references, supporting supervised enhancement. The unpaired subset comprises real reef and marine-habitat scenes that better reflect natural underwater variability.

The dataset contains extracted and labeled coral images which show different coral species and substrate types. The dataset includes multiple coral categories which consist of branching corals massive corals brain corals plate corals and background classes that include sand and algae. The dataset divides into three parts which include training 70 percent validation 15 percent and testing 15 percent to ensure both statistical reliability and unbiased assessment. The training set establishes model parameters while the validation set helps optimize hyperparameters and stop overfitting. The testing set maintains exclusive use for complete testing of final performance results. The split provides a method to conduct fair comparisons against all current leading-edge methods.

3.3 Underwater Image Enhancement Module

Underwater images suffer severe wavelength-selective attenuation: red light is absorbed within a few meters while blue and green persist, producing the characteristic blue–green cast and a reduction in

effective contrast. Scattering superimposes a veiling haze that blurs object boundaries. Classifying such images directly is unreliable because the extracted features are dominated by degradation rather than by coral morphology. An enhancement module is therefore applied as the first stage of the pipeline.
Let I_r represent the raw underwater image and I_e denote the enhanced image. The enhancement function can be expressed as $I_e = fI_r$ where $f(.)$ is a composite enhancement operation. The enhancement procedure includes three main components which involve white balance correction to eliminate colour casts and contrast stretching for expanding dynamic range and Retinex-based illumination normalization to solve problems with uneven lighting. The deep learning restoration network provides an additional option which enables users to learn complex image mappings between degraded images and high-quality images. The network restores visibility through the process of recovering structural details and chromatic details which underwater acquisition lost. The enhanced images show better sharpness and improved colour accuracy and higher contrast which helps the feature extraction process to identify important coral characteristics. The proposed system uses enhancement in the pipeline to reduce noise and distortion which leads to better classification accuracy for the classifier.

3.4 Preprocessing and Data Augmentation

Following enhancement, all images pass through a preprocessing stage that standardises input dimensions and reduces computational load. Images are resized to 224 × 224 pixels, matching the input resolution of the convolutional backbones, and pixel intensities are normalised to the range [0, 1] to stabilise gradient updates. Median filtering or Gaussian smoothing removes residual speckle noise and improves textural uniformity. Data augmentation then increases sample diversity and guards against overfitting: horizontal and vertical flips, rotation, random cropping, zoom, and brightness and contrast jitter. These transformations emulate realistic variation in camera pose, depth and ambient illumination, improving the model's ability to generalise to unseen reef imagery.

3.5 Deep Feature Extraction

Handcrafted descriptors such as colour histograms, Local Binary Patterns (LBP) and texture statistics capture only part of the morphological variance that distinguishes coral genera. The proposed approach instead relies on automatic deep feature learning with convolutional neural networks, which extract hierarchical representations directly from raw pixel values through stacked convolutional layers. The feature extraction stage is expressed as F = CNN(I_e), where I_e is the enhanced image and F the resulting feature map.
The learned feature maps are represented by the component F. The early layers of the system identify fundamental elements through their ability to detect edges and textures. The system uses its three-layer structure, which contains three different levels of processing, to identify basic visual components through its first level and to establish advanced semantic relationships through its last level, which defines coral types. The model uses its hierarchical structure to identify tiny differences in coral structure, which manual methods fail to detect.

3.6 Evaluated Architectures

The enhanced and preprocessed images are passed to a deep convolutional classifier comprising stacked convolutional layers with rectified linear unit (ReLU) activations, max-pooling layers and fully connected layers. Transfer learning is used throughout: ImageNet-pretrained ResNet, MobileNet and EfficientNet backbones supply general visual priors, and each network is fine-tuned on the coral dataset to adapt it to the underwater domain. The final classification layer applies a softmax function to produce class probabilities:

$$y = softmax(WF + b) \quad \text{Eqn (1)}$$

Eqn (1) represents the final classification layer of a neural network. Here W and b is representing weights and biases, respectively while F represents vector feature extracted from previous layer. The linear transformation WF+B produces a raw score called Logits. The softmax function then coverts these logits into probabilities using

$$softmax(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad \text{Eqn (2)}$$

The above eqn (2) produces outputs which range from 0 to 1 and their total equals 1. The y output becomes a probability vector which shows the predicted class through its highest value.

3.7 Training Strategy and Optimization

The network is trained by supervised learning on labelled coral imagery. At each iteration a forward pass generates predictions and the cross-entropy loss quantifies the discrepancy against the ground-truth labels; backpropagation with gradient descent then updates the weights. The Adam optimizer is used for its adaptive learning-rate behavior and rapid convergence. Training uses a batch size of 32, an initial learning rate of 0.001 and a budget of 100 epochs, with early stopping on validation loss to prevent overfitting. Dropout regularization and batch normalization further improve generalization and stabilize training.

Model selection and meta-feature generation follow a stratified 13-fold cross-validation protocol; the out-of-fold construction of the meta-training set is described in Section 4.10.2. The held-out test partition is excluded from every fold and is used once, for final reporting only.

3.8 Evaluation Metrics

The trained models are evaluated with a complementary set of metrics. Accuracy measures overall correctness, precision quantifies the reliability of positive predictions, and recall measures the proportion of true instances retrieved. The confusion matrix exposes per-class behaviour and systematic misclassification patterns, while Receiver Operating Characteristic (ROC) curves and the associated Area Under the Curve (AUC) scores characterise class separability independently of any single decision threshold.

Accuracy is computed as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Eqn (3)}$$

In Eqn (3), TP stands for true positives while TN denotes true negatives and FP stands for false positives and FN indicates false negatives. The metrics deliver complete insight into how well the classification system performs its tasks.

3.9 Comparative Protocol

Each architecture is evaluated under a common protocol so that differences in reported performance reflect architectural capacity rather than experimental setup. Within each partitioning scheme the dataset, class definitions, augmentation policy, input resolution and metric definitions are held constant, and every model is initialised from ImageNet-pretrained weights. Comparison proceeds on three levels: aggregate performance across all eight classes; per-class precision, recall and F1-score, which expose failure modes that aggregate figures conceal; and the correlation structure of the residual errors across architectures, which determines whether multi-model approaches can be expected to yield further gains.

3.10 Visualization and Interpretability

Grad-CAM heatmaps are used to interpret model decisions by visualizing the image regions that most influenced each prediction. The resulting maps concentrate on coral structures rather than on background substrate or water column, indicating that the network bases its predictions on biologically meaningful evidence. Such visual explanations are essential to the transparency and trustworthiness of automated underwater monitoring systems.

4 Experimental Results

Table 1 Per-class precision, recall, F1-score and support for seven transfer-learning backbones on the eight-class coral dataset under a fixed train/validation/test partition.

Model	Coral Reef Classes	Precision	Recall	F1 – Score	Support	Accuracy
EfficientNetB0	Acropora corals	1.00	1.00	1.00	42	0.81
	Bleached Corals	0.74	0.60	0.67	43	
	Branching Soft Coral	0.66	0.64	0.65	42	
	Dead Corals	0.78	0.87	0.82	45	
	Foliose Corals	0.96	1.00	0.98	43	
	Healthy Corals	0.63	0.71	0.67	48	
	Lobe Coral	0.65	0.59	0.62	22	

Model	Coral Reef Classes	Precision	Recall	F1 – Score	Support	Accuracy
	Staghorn Corals	1.00	0.95	0.98	43	
ResNet50	Acropora corals	0.96	0.94	0.95	87	0.76
	Bleached Corals	0.81	0.66	0.73	83	
	Branching Soft Coral	0.45	0.84	0.59	83	
	Dead Corals	0.74	0.92	0.82	89	
	Foliose Corals	1.00	1.00	1.00	80	
	Healthy Corals	0.62	0.24	0.35	103	
	Lobe Coral	0.67	0.47	0.55	43	
	Staghorn Corals	1.00	1.00	1.00	82	
ResNet101	Acropora corals	0.96	0.76	0.85	84	0.67
	Bleached Corals	0.61	0.69	0.65	85	
	Branching Soft Coral	0.38	0.28	0.32	82	
	Dead Corals	0.60	0.56	0.58	89	
	Foliose Corals	0.98	0.94	0.96	85	
	Healthy Corals	0.55	0.62	0.58	95	
	Lobe Coral	0.20	0.31	0.24	42	
	Staghorn Corals	0.97	0.98	0.97	85	
ResNet152V2	Acropora corals	0.93	0.95	0.94	42	0.72
	Bleached Corals	0.71	0.58	0.64	43	
	Branching Soft Coral	0.53	0.50	0.51	42	
	Dead Corals	0.61	0.62	0.62	45	
	Foliose Corals	0.98	0.98	0.98	43	
	Healthy Corals	0.60	0.58	0.59	48	
	Lobe Coral	0.33	0.45	0.38	22	
	Staghorn Corals	0.98	1.00	0.99	43	
VGG16	Acropora corals	1.00	0.98	0.99	43	0.78
	Bleached Corals	0.73	0.56	0.63	43	
	Branching Soft Coral	0.68	0.45	0.54	42	
	Dead Corals	0.66	0.87	0.75	45	
	Foliose Corals	0.95	0.98	0.97	43	
	Healthy Corals	0.64	0.77	0.70	48	
	Lobe Coral	0.55	0.50	0.52	22	
	Staghorn Corals	0.93	0.98	0.95	43	
VGG19	Acropora corals	0.98	0.76	0.86	84	0.65
	Bleached Corals	0.64	0.61	0.63	85	
	Branching Soft Coral	0.28	0.13	0.18	82	
	Dead Corals	0.62	0.52	0.56	89	
	Foliose Corals	0.89	1.00	0.94	85	
	Healthy Corals	0.48	0.72	0.57	95	

Model	Coral Reef Classes	Precision	Recall	F1 – Score	Support	Accuracy
	Lobe Coral	0.26	0.48	0.34	42	
	Staghorn Corals	1.00	0.88	0.94	85	
Xception	Acropora corals	1.00	1.00	1.00	42	0.79
	Bleached Corals	0.79	0.60	0.68	43	
	Branching Soft Coral	0.49	0.76	0.60	41	
	Dead Corals	0.77	0.89	0.82	45	
	Foliose Corals	1.00	1.00	1.00	43	
	Healthy Corals	0.71	0.42	0.53	48	
	Lobe Coral	0.62	0.62	0.62	21	
	Staghorn Corals	0.98	1.00	0.99	42	

4.1 Experimental Configuration

All seven backbones were benchmarked under an identical protocol so that differences in reported performance are attributable to architecture rather than to data handling or optimisation. The curated dataset of 3,250 images was partitioned class-wise into 2,598 training, 324 validation and 328 test images, preserving the eight-class balance in every partition; the test partition was held out and used only once, for final reporting.

Every backbone was initialised with ImageNet-pretrained weights. Input images were resized to 224 × 224 pixels and normalised using the statistics expected by each pretrained network. The convolutional base was frozen and a common classification head — global average pooling, dropout, and a dense softmax layer over the eight classes — was attached, so that the number of trainable parameters differs only through the width of each backbone's final feature map. Training used the Adam optimiser with categorical cross-entropy loss, a batch size of 32 and a maximum of 100 epochs, with early stopping on validation loss and learning-rate reduction on plateau.

Data augmentation during training comprised random horizontal and vertical flips, rotation, zoom and brightness jitter, together with the underwater-specific perturbations described in Section 3.4. No augmentation was applied at validation or test time. The baseline benchmark reported in this section was implemented in [FRAMEWORK — confirm] and executed on a single GPU. The proposed framework of Section 4.10 is implemented separately in PyTorch with PyTorch Lightning and the timm model library, with random seeds fixed at 42 for PyTorch, NumPy and CUDA and cuDNN placed in deterministic mode.

4.2 EfficientNetB0

We have pre-trained and fine-tuned EfficientNetB0, which is a deep learning model, to identify all eight coral classes. The Dataset contains 2598 images for training, 324 images for validation, 328 test images, which were divided evenly among all eight classes. The EfficientNetB0 feature extractor consists of 4,049,571 frozen parameters, which connects to global average pooling and dropout and to a dense output layer that has 10,248 trainable parameters.

Training Metrics (100 Epochs)

The training loss fell from 1.50 to 0.31 while the accuracy increased from 0.44 to 0.87. The validation accuracy reached its highest point at 0.83 before finishing at 0.80 while the validation loss decreased from 0.88 to 0.47.

Accuracy and Loss Formulas

Accuracy (Acc):

$$Acc = \frac{\text{Number of correct predictions}}{\text{Total predictions}} \quad \text{Eqn (4)}$$

Here, in Eqn (4), number of correct predictions includes all samples where the predicted label matches the true label (both true positives and true negatives). Total predictions are the total number of test samples.

The calculation produces a value range which extends from 0 to 1, with higher accuracy values showing superior performance. However, accuracy alone may be misleading when classes are imbalanced, so metrics like precision, recall, and F1-score are also important.

Cross-entropy loss ($\mathcal{L}$):

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y(i,c)\, log\{p_{(i,c)}\} \quad \text{Eqn (5)}$$

Above eqn (5) produces the negative log likelihood loss represents the bathymetric Cross which is the main training objective for CNN-based underwater coral classification system. The total number of training images is represented by N while C denotes the count of different coral classes. Where $y_{i,c}$is the true label (one-hot encoded), $p_{i,c}$is the predicted probability for class c.

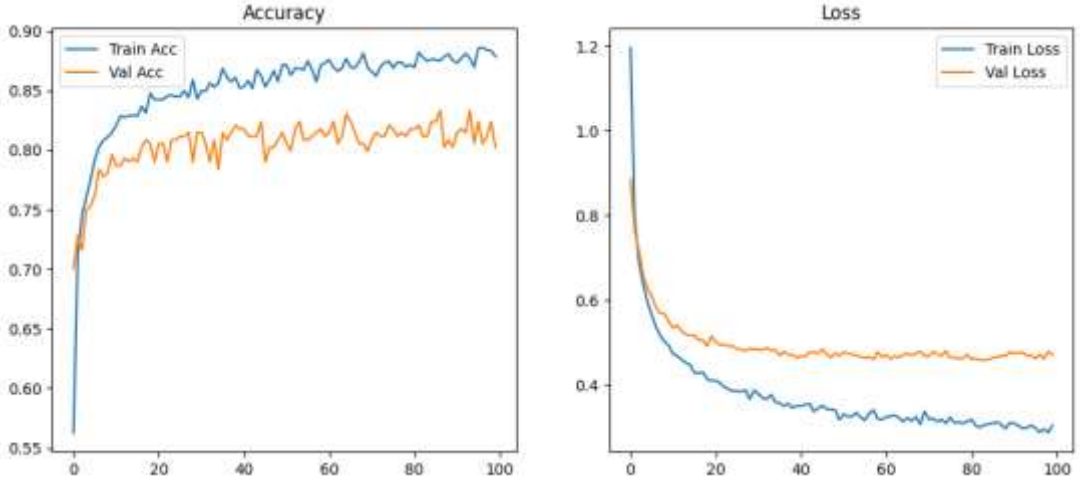
In this work, the loss function directs the convolutional neural network to enhance its ability to identify coral species through weight modifications during backpropagation. The loss decreases when predictions correctly identify the true coral labels but increases when predictions fail or show uncertainty because of underwater noise and low contrast and colour distortion. The underwater coral detection model achieves better classification accuracy through loss minimization, which directly enhances its performance.

Test Results and Evaluation

Table 2 Per-class test performance of the EfficientNetB0 baseline (328 test images).

Class	Precision	Recall	F1-score	Support
Acropora corals	1.00	1.00	1.00	42
Bleached Corals	0.74	0.60	0.67	43
Branching Soft Coral	0.66	0.64	0.65	42
Dead Corals	0.78	0.87	0.82	45
Foliose Corals	0.96	1.00	0.98	43
Healthy Corals	0.63	0.71	0.67	48
Lobe Coral	0.65	0.59	0.62	22
Staghorn Corals	1.00	0.95	0.98	43
Overall Accuracy			0.81	328

Fig. 1 Confusion matrix and ROC curves for the EfficientNetB0 baseline.



Classification Metrics Formulas

Precision (P):

The performance metric evaluates the accuracy of the classification model for predicting the positive class eqn(6). Precision measures the proportion of correctly predicted positive samples (True Positives, TP) out of all samples that the model predicted as positive, which includes both correct predictions (TP) and incorrect positive predictions (False Positives, FP)

$$P = \frac{TP}{TP + FP} \qquad \text{Eqn (6)}$$

Recall (R):

$$R = \frac{TP}{TP + FN} \qquad \text{Eqn (7)}$$

The metric known as recall eqn(7) which scientists refer to as sensitivity and which others call the true positive rate measures the performance of a classification system in identifying all actual positive results. The system calculates the True Positive Rate as the number of correctly identified positive results (True Positives) divided by the total count of actual positive results which consists of both identified positive results and unrecognized positive results (False Negatives).

F1-score:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \qquad \text{Eqn (8)}$$

The F1 score eqn(8) functions as a performance metric which measures the balance between precision and recall through its combined assessment. The F1 score establishes a balance between both metrics because it evaluates the positive prediction accuracy through precision and actual positive detection through recall. The formula calculates F1 score through its mathematical calculation which uses precision and recall values to determine their harmonic mean instead of using their arithmetic mean. The model needs to show good performance in all areas because the assessment requires both sections to be passing. The F1 score summarizes both requirements into one value which makes it highly effective for imbalanced datasets and situations where both false positives and false negatives need to be assessed. The score ranges from 0 to 1 with 1 representing complete accuracy in classification tasks.

Confusion Matrix element (C_{ij}): Number of samples of true class i predicted as class j.

4.3 ResNet50

The researchers used a deep convolutional neural network which depends on the ResNet50 architecture to identify multiple coral species through their underwater images. For my research work I have collected 3,250 coral images which researchers divided into eight separate categories that included Acropora Corals, Bleached Corals, Branching Soft Coral, Dead Corals, Foliose Corals, Healthy Corals, Lobe Coral, and Staghorn Corals. The dataset provided 2,600 samples for training purposes while 650 images remained for validation testing to assess how well the model could generalize its results.

All images were resized to 224 × 224 × 3 dimensions because that size matched the requirements of ResNet50 input dimensions. The pretrained ResNet50 network which used ImageNet weights as its starting point functioned as the backbone for feature extraction after its original fully connected classification layers were deleted. The backbone system used a Global Average Pooling (GAP) layer which connected to a Dropout layer and then to a Dense output layer that contained eight neurons which used softmax activation for classification of eight coral categories. The architecture used in this system contained 23.6 million parameters of which 19.47 million parameters could be trained while 4.13 million parameters stayed inactive during the fine-tuning process.

The training process used 100 epochs to train the model while using Adam optimizer together with categorical cross-entropy loss. The researchers chose a batch size of 32 because it fit within the GPU execution limits of Google Colab. The Keras Image Data Generator interface handled data feeding operations while standard preprocessing methods were applied without using special underwater augmentation techniques. The researchers evaluated model performance after each epoch by measuring validation accuracy and loss results.

Quantitative Evaluation

The model achieved a training accuracy of 97.05% after completing 100 training epochs because it successfully learned features from the training data until it reached almost complete training data convergence. The system demonstrated partial success in processing new samples because its validation accuracy reached 76.31%. The assessment of each class reveals different levels of success in classifying different classes: Acropora Corals achieved a precision of 0.96 and recall of 0.94 which resulted in an F1-score of 0.95 across 87 validation samples that showed high certainty in their identification.

The Bleached Corals achieved a precision of 0.81 and a recall of 0.66 which produced an F1-score of 0.73 because the system mistakenly identified this coral state as different states that shared similar spectral characteristics.

The Branching Soft Coral reached a recall value of 0.84 but its precision score remained below 0.45 which produced an F1-score of 0.59. The Dead Corals assessment produced excellent results because they achieved an F1-score of 0.82 and a recall rate of 0.92 across 89 samples. The Foliose Corals study produced perfect classification results because all 80 samples received correct identification through the study. The Healthy Corals study showed a precision rate of 0.62 and a recall rate of 0.24 which produced an F1 score of 0.35. Lobe Coral achieved moderate recognition performance (precision = 0.67, recall = 0.47, F1 = 0.55). The Staghorn Corals group obtained assessment results that showed perfect precision and recall and F1-score performance results of 1.00. The validation accuracy of the model reached 0.76 while both macro and weighted-average F1 scores reached 0.75. The ResNet50-based framework effectively identifies various coral categories despite its decreased performance with visually similar and naturally variable classes.

4.4 ResNet101

The evaluation results of the ResNet101 model were measured through established multi-class evaluation metrics using eqn (9). Precision for class i measures the proportion of correctly predicted samples of class of all samples predicted to be class i:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \quad \text{Eqn (9)}$$

Where TP_i is true positives, and FP_i is false positives for class i.

Recall (Sensitivity) for class i captures the proportion of correctly predicted samples of class $i$out of all actual samples of class i:

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \quad \text{Eqn (10)}$$

Where FN_i is false negatives for class i.

F1-score for class $i$is the harmonic mean of precision and recall:

$$\text{F1}_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad \text{Eqn (11)}$$

Accuracy: The overall proportion of correctly classified samples:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad \text{Eqn (12)}$$

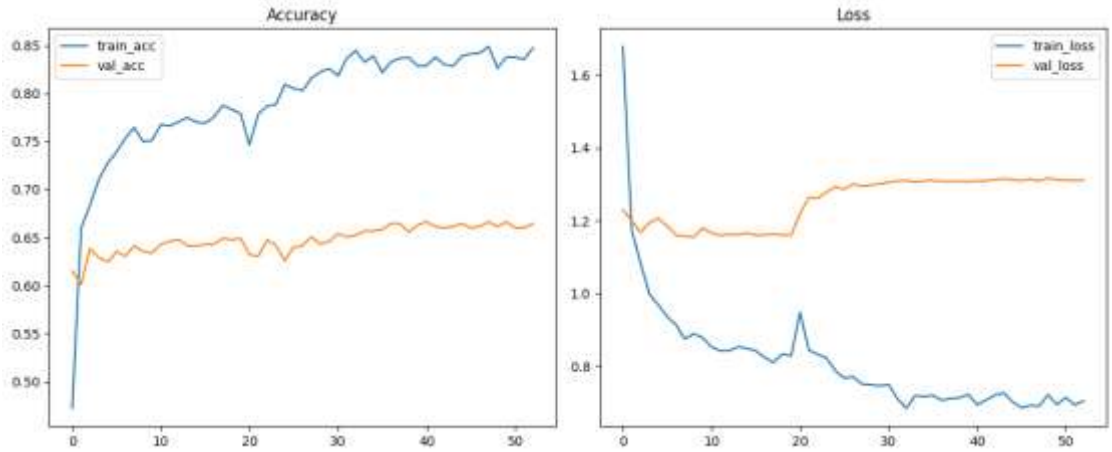
Macro Average: The Macro Average eqn (13) calculates the overall performance of a multi-class classification model by first computing a metric (such as precision, recall, or F1-score) individually for each class, and then taking the simple arithmetic mean across all classes. The total number of classes N includes all categories which exist in the example of different coral types such as healthy coral and bleached coral and algae-covered coral and sand and rocks. Macro averaging treats all classes with equal weight because it does not consider the different sample sizes that exist in each class.

$$\text{Macro Average} = \frac{1}{N}\sum_{i=1}^{N}(Metric)_i \quad \text{Eqn (13)}$$

Weighted Average: The Weighted Average eqn (14) of performance metrics which include precision and recall and F1-score in multiparty classification. The weighted average method treats each class with different value according to its support which determines class importance through actual sample count of

the i-th class. The formula uses (Metric)_i which represents class metric values as N stands for total class number. The formula calculates class metrics by multiplying them with their respective sample counts which results in total metric values that get divided by overall sample size to create a metric that gives more weight to larger classes.

$$\text{Weighted Avg} = \frac{\sum_{i=1}^{N} 1/n \, \text{Support}_i \times \text{Metric}_i}{\sum_{i=1}^{N} 1/n \, \text{Support}_i}$$



4.5 ResNet152V2

The ResNet152V2-based coral classifier's performance evaluation used multi-class assessment metrics to test its accuracy across eight different classes. The study uses standard classification metrics to assess model performance which is necessary for both reporting and making comparisons.

Key Evaluation Metrics and Formulas

Precision (P) refers to the ratio of positive instances that were predicted correct (i.e. were predicted to be of that class and were actually of that class) to the total number of instances predicted to be in the class.

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

Recall (R): Can accurately test as to whether all positive samples can be identified. Eqn (15)

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

Eqn (16)

where TP is True Positives and FN is False Negatives.

F1-Score: Balances precision and recall:

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

Eqn (17)

Accuracy: The classification accuracy eqn (18) shows which percentage of correct predictions a model made across all its classes. Here, C denotes the total number of classes, TP_i indicates the number of correctly predicted samples for the i-th class (true positives for each class), and N is the total number of samples in the dataset. The total number of correct predictions emerges from summing all true positives across different classes while the correct prediction rate results from dividing that total by all samples.

$$Accuracy = \frac{\sum_{i=1}^{C} TP_i}{N}$$

Eqn (18)

Macro Average: Eqn (19) shows the Macro Average of evaluation metric M which includes precision and recall and F1-score for multi-class classification evaluations. The total number of classes is represented by C while Mi shows the metric value which was calculated for the i-th class. The formula simply adds the metric values of all classes and divides by the number of classes, giving the simple arithmetic mean. The system treats all classes as having the same weight because it treats all classes as having the same importance to the calculation.

Eqn (19)

$$Macro\ Avg(M) = \frac{1}{C}\sum_{i=1}^{C} M_i \qquad \text{Eqn (19)}$$

Weighted Average: The equation (20) establishes the weighted average performance metric which operates across all classes in a multi-class classification system. The variable C denotes the complete count of classes in the system while Mi represents the assessment metric value which includes Precision and Recall and F1-score metrics for the $i\ th$ class. The term N=∑i=1C Support iN=∑i=1C represents the complete sample count within the dataset. The system assigns greater value to classes which have higher sample counts through the process of multiplying each class metric by its support and then calculating the total weighted values which gets divided by the overall sample count.

$$WeightedAvg(M) = \frac{1}{N}\sum_{i=1}^{C} Support_i \,.\, M_i$$

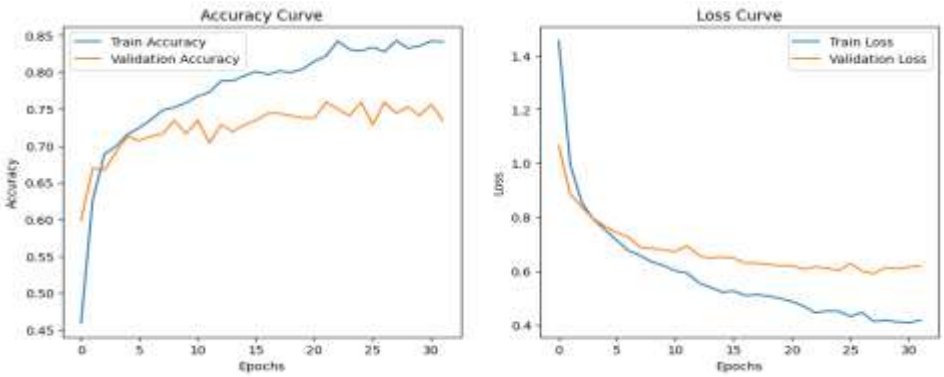
The confusion matrix displays the actual and incorrect prediction results for every category which enables the detection of incorrect predictions together with their pattern of occurrence.

ROC and AUC: ROC curves per class show separability; AUC values $(0.91 - 1.00)$ confirm strong discrimination for most classes:

$$AUC = \int_0^1 TPR(FPR)d(FPR)$$

Training curves of accuracy and loss across different epochs manifest the lessons of co generalization gaps. Eqn (21)

Fig. 2 Training and validation accuracy and loss curves for ResNet152V2.



4.6 VGG16

The VGG16 model was trained and tested on the underwater coral dataset using a stratified split across the eight coral categories. Training accuracy rises from approximately 0.15 at initialisation to above 0.75, while validation accuracy stabilises at approximately 0.74 by the end of training. The steady decline in loss alongside rising accuracy indicates effective convergence, and the narrow gap between the training and validation curves indicates limited overfitting.

Class-wise Insights

The visual characteristics of Acropora and Foliose and Staghorn corals (F1 > 0.95) provide evidence that these categories can be recognized by the model. The Dead and Healthy corals show moderate performance because their identification accuracy depends on the balance between recall and precision which exists to some degree across related categories. Branching Soft and Bleached and Lobe Coral show low scores (F1 < 0.6) because of three reasons: intra-class variation and class imbalance and insufficient representation in the feature space which creates chances for targeted data augmentation to solve this issue.

Reliability and Robustness

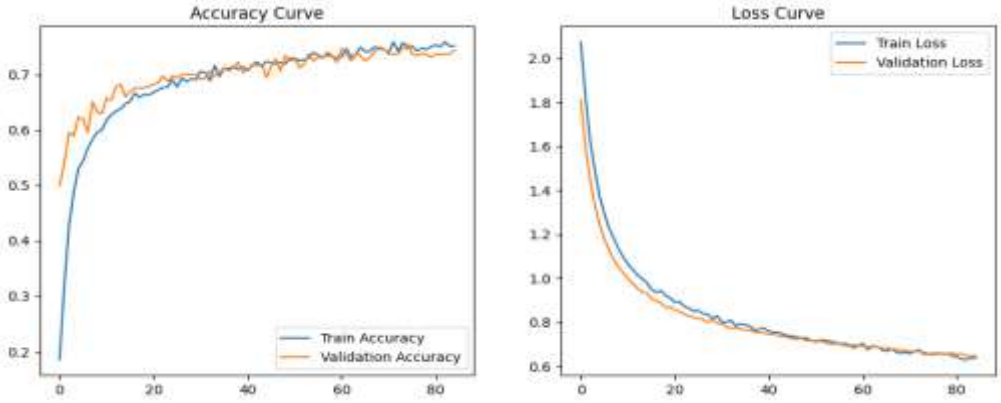
The macro average measure achieved an F1 score of 0.76 which assesses model performance across all classes without considering their sample distribution and demonstrates equal performance across multiple classes. The weighted average achieved an F1 score of 0.77 which assesses operational accuracy for coral

survey work by considering the distribution of examples across different classes. The overall accuracy of the model reached 0.78 which demonstrated its capacity to predict unseen data in stratified split testing.

Comparative Methodological Notes

The VGG16 backbone achieves competitive performance on this dataset through its transfer learning abilities and fine-tuning of its classification head layers despite having lower parameter requirements than ResNet152V2 which uses deeper architecture. The training epochs demonstrate improvements which indicate that the selected learning rate and optimization schedule work well with the dataset size. The combination of advanced regularization techniques through dropout and batch normalization with stratified data splitting methods protects against overfitting to maintain high generalization performance.

Fig. 3 Training and validation curves and confusion matrix for VGG16.



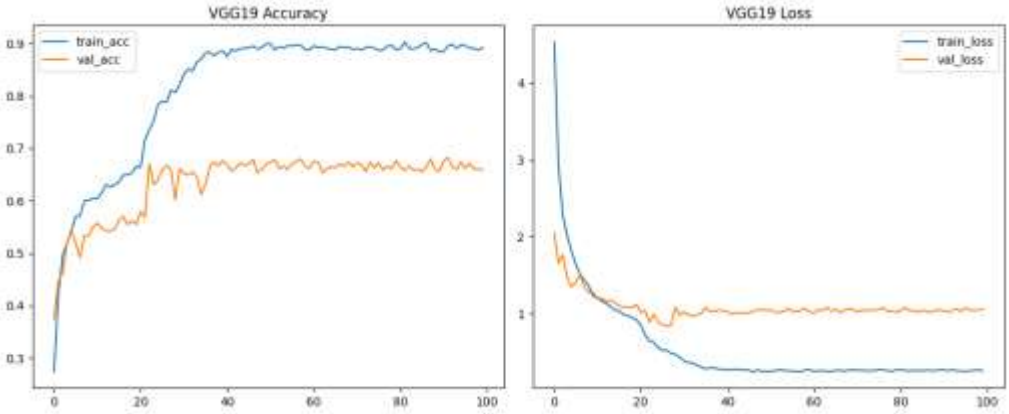
4.7 VGG19

Dataset and Training Protocol

The dataset contains 2603 training images and 647 validation images to detect 8 different coral species. The training process used stratified splitting and class weight calculations to handle class imbalances which resulted in different Lobe Coral weight of 1.90 and other classes weight of approximately 0.85 to 0.98. VGG19 serves as the backbone model which operates under pre-trained ImageNet weights and includes a dedicated classification module that uses Global Average Pooling and two Dense layers which contain 512 and 8 units and Dropout regularization. The complete system consists of 20291144 total parameters which include 102M parameters that can be updated during training.

The initial training process began with training the top layers. The model achieved an accuracy boost from about 0.21 to above 0.68 during its top performance periods. The model proved its backbone system value through its ability to decrease losses from 5.38 to approximately 0.94 while extracting essential features. The process involves fine-tuning the last block of the convolutional network. The validation accuracy increased from about 0.57 to its highest point of about 0.68 during the early training sessions. The training process used progressive learning rate reduction through ReduceLROnPlateau which resulted in a final accuracy of approximately 0.90 that proved the model could generalize to new data.

Fig. 4 Training and validation curves for VGG19 during head training and fine-tuning.



4.8 Xception

The researchers assessed the proposed underwater coral classification system by using Xception deep convolutional neural network as the primary classifier because it delivered fast performance and advanced feature extraction capabilities. Xception uses depth wise separable convolutions to separate its feature extraction process into two distinct stages which decreases its required parameters while maintaining strong representational power. The output feature map of a standard convolution process computes its results through the following equation.

$$y_{i,j,k} = \sum_{m=1}^{M} \sum_{u=1}^{K} \sum_{v=1}^{K} W_{u,v,m,k} \cdot X_{i+u,j+v,m}\, 1$$

where X denotes the input tensor, W the kernel weights, and K the kernel size. The main prc of this proposed method incurs a major computational cost of $K^2.M.N.H.W$ Eqn (25)

Depth-wise convolution is used for the replacement for Xception decomposes process,

$$Z_{i,j,m} = \sum_{u=1}^{K} \sum_{v=1}^{K} D_{u,v,m} \cdot X_{i+u,j+v,m1}$$

which independently filters each channel, followed by a pointwise 1×1 convolution Eqn (26)

$Y_{i,j,k} = \sum_{m=1}^{M} P_{m,k}.Z_{i,j,m}$ is used for the combined inter-channel date for the processing. This reduces complexity to $(K^2.M + M.N).H.w_1$. The process demonstrates faster convergence together with improved generalization performance through its execution on lowercase data which reduces standard convolution operations.

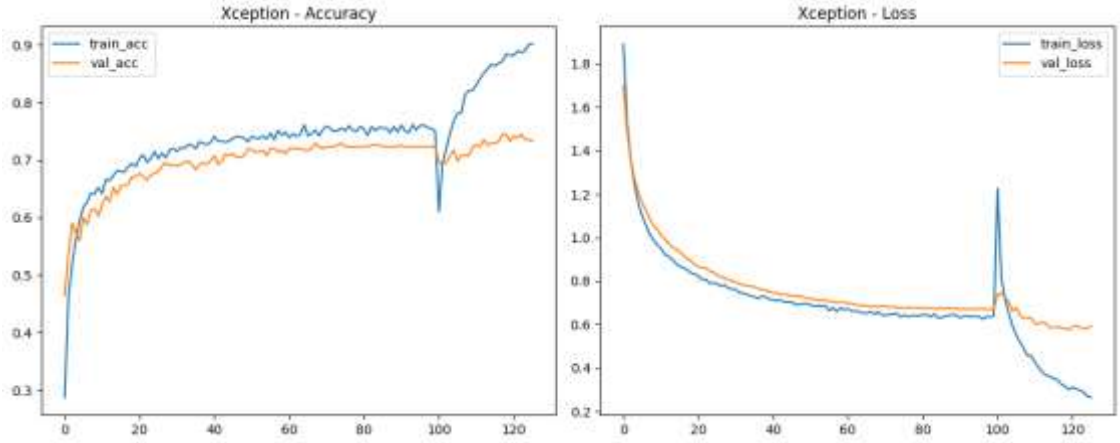
The learning curves demonstrate that the training process developed stable optimization capabilities which enabled successful feature development. The model achieved an initial accuracy rate between 30 and 45 percent which began to increase at a rapid pace after it learned to recognize fundamental edge and texture patterns. The accuracy reached its final training value of 90 to 92 percent after 40 to 60 epochs while the validation accuracy stabilized between 73 to 75 percent. The learning curves demonstrate that the training process acquired stable optimization abilities which resulted in successful feature acquisition. The model achieved an initial accuracy of 30 to 45 percent which increased at a fast rate after it acquired the ability to recognize basic edge and texture elements. The accuracy reached its final training value of 90 to 92 percent after 40 to 60 epochs while the validation accuracy stabilized between 73 to 75 percent. Classification accuracy was computed using

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

The research successfully identified most coral samples through its identification methods. The loss curves demonstrated continuous loss decline which resulted in training loss decreasing from 1.9 to 0.25 and validation loss reaching a point of 0.6. The optimization procedure employed categorical cross-entropy loss which researchers defined as $L = -\sum_{c=1}^{C} y_c\, log(y_c)$, where y_(c)represents the actual result and denotes the estimated probability. The model achieved successful classification error reduction thro process, which resulted in the development of distinct coral morphological identification Eqn (27)

function experienced a temporary increase during the later training epochs, which researchers traced back to the impact of their learning rate modifications and the inherent variation present in their mini-batch samples, yet the system achieved stable operational performance through its training structure. The slight gap between training and validation accuracies indicates mild overfitting which is expected due to the inherent variability turbidity and illumination inconsistencies in underwater imagery.

Fig. 5 Training and validation accuracy and loss curves for Xception.



Validation Results

ROC Analysis

The system demonstrates its ability to classify most coral categories based on their AUC results which range from 0.87 to 1.00. The system demonstrates high classification accuracy because both Micro AUC and Macro AUC scores exceed 0.93.

Training and Validation Curves

The model achieved a train accuracy which exceeded 0.89 while the validation accuracy remained within the range of 0.65 to 0.7 after we completed the fine-tuning process. The train loss decreased steadily until it reached approximately 0.25 while the validation loss decreased until it reached approximately 1.02. The model demonstrates minimal overfitting because it shows only a slight difference between training and validation performance which is confirmed by its continuous validation results.

Confusion Matrix

Most coral classes achieve high correct classification rates. The classes which include Acropora and Foliose and Healthy and Staghorn show no confusion between their respective identities. The off-diagonal values of Bleached and Branching Soft Coral remain high because visual similarities between samples together with sample imbalance lead to ongoing classification errors.

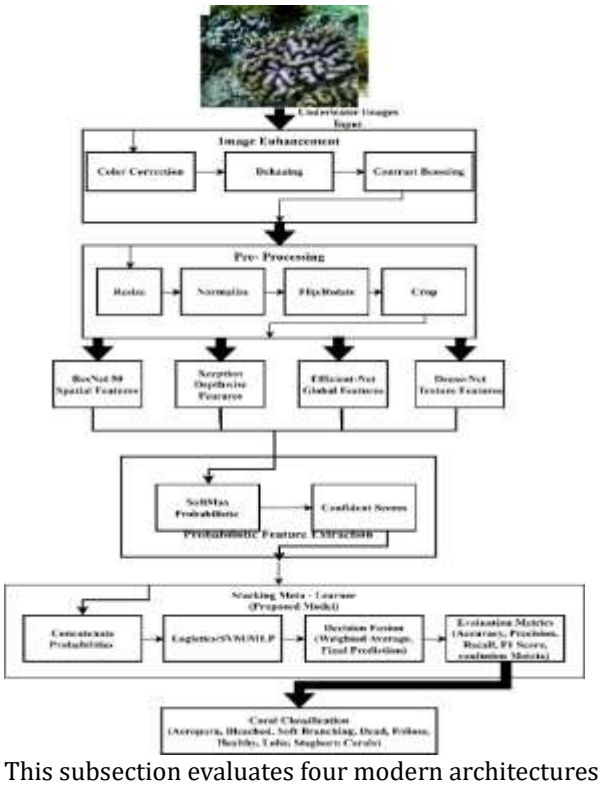
4.9 Summary of the Fixed-Partition Evaluation

Across the seven backbones, overall accuracy ranges from 0.65 (VGG19) to 0.81 (EfficientNetB0), with Xception (0.79), VGG16 (0.78), ResNet50 (0.76), ResNet152V2 (0.72) and ResNet101 (0.67) in between. Two patterns are consistent across every architecture. First, the morphologically distinctive classes — Acropora, Foliose and Staghorn — are recovered almost perfectly, with F1-scores above 0.94 in nearly all cases. Second, every backbone degrades sharply on the same three classes: Branching Soft Coral, Bleached Corals and Lobe Coral, whose F1-scores fall below 0.6 for most models and as low as 0.18 for VGG19. Critically, the models do not fail identically on these classes: ResNet50 attains 0.84 recall on Branching Soft Coral where VGG19 attains 0.13, while VGG16 reaches 0.77 recall on Healthy Corals where ResNet50 reaches only 0.24. This complementarity of errors — rather than any single architecture's headline accuracy — is the property the proposed ensemble is designed to exploit.

4.10 Cross-Validated Evaluation of Modern Architectures

4.10.1 Study Design

Fig. 6 Overall architecture of the proposed METASTACK-NET framework.



This subsection evaluates four modern architectures under a single, uniformly applied protocol so that the comparison isolates architectural capacity from experimental variation. Two convolutional families, one

efficiency-optimised convolutional network and one vision transformer are included, chosen to span the design space rather than to sample it exhaustively. The evaluation proceeds in four stages: dataset preparation and augmentation; independent training of each architecture under stratified cross-validation; aggregation of out-of-fold predictions; and comparative analysis of aggregate and per-class performance. A preliminary ensemble formulation over the four architectures is set out in Section 4.10.3 and is reported as exploratory only.

All coral images undergo initial processing which includes resizing to a standard size and normalization to establish uniformity. The research team applies data augmentation techniques that include horizontal flipping and rotation and random cropping and colour jittering to reduce overfitting while creating artificial underwater conditions. Let the training dataset be defined as

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

where $x_i$represents the input image and $y_i \in \{1, \dots, C\}$corresponds to one of $C = 8$coral classes.

4.10.2 Evaluated Architectures and Training Protocol

Eqn (28)

The base learner pool comprises four architecturally distinct networks, each initialised fr pretrained weights and fine-tuned independently: ResNet50, which learns residual features; EfficientNet-B3, a compound-scaled convolutional network offering high accuracy at low parameter cost; ViT-Base, a vision transformer that models global dependencies through self-attention; and DenseNet121, whose dense connectivity promotes feature reuse and gradient flow. All four are instantiated through the timm library with a dropout rate of 0.4 applied before the classification head.

Each base learner f_k maps an input image to a class-posterior vector through a learned nonlinear transformation:

$$f_k(x; \theta_k) \rightarrow p_k$$

where $k \in \{1,2,3,4\}$, $\theta_k$denotes learnable parameters, and $p_k \in R^C$is the SoftMax probability vector.

Training uses categorical cross-entropy loss:

$$\mathcal{L}_{CE} = -\sum_{c=1}^{C} y_{ic}\, log(\hat{y}_{ic})$$

Eqn (30)

where $\hat{y}_{ic}$is the predicted probability for class c.

Base learners are trained under stratified 13-fold cross-validation with shuffling and a fixed random seed of 42. Stratification preserves the eight-class proportions within every fold, which matters because the minority coral classes would otherwise be unevenly represented across partitions. Each fold trains on twelve partitions and predicts the held-out partition; the reported base-learner metrics in Section 5.1 are computed on the concatenation of these held-out predictions across all thirteen folds, so every sample contributes exactly one prediction made by a model that did not see it during training. Optimisation uses AdamW with a learning rate of 5×10^{-5}, weight decay 1×10^{-3}, batch size 16 and a maximum of 50 epochs per fold, with early stopping on validation loss after seven epochs without improvement.

4.10.3 Exploratory Ensemble Formulation

The per-class analysis in Section 5.2 shows that the four architectures do not fail on the same images, which raises the question of whether their outputs can be combined productively. This subsection sets out a stacking formulation for that purpose. It is presented as an exploratory extension rather than as a result of this study: the evaluation reported for it is in-sample, for the reasons given in Section 5.4, and it is therefore excluded from the comparative claims made in Sections 4 to 5.2. The class-posterior vectors of the four architectures are concatenated into a meta-feature vector:

$$z_i = [p_1 \oplus p_2 \oplus p_3 \oplus p_4] \in R^{4C}$$

For C = 8 classes and four base learners this yields a 32-dimensional meta-feature space in which each axis carries one model's belief about one class, and in which disagreement between models is directly representable. The meta-feature vector is passed to a meta-learner g(·), implemented as a fully connected classifier with a single hidden layer of 512 units, ReLU activation, batch normalisation and dropout of 0.5, projecting to the eight output classes:

$$\hat{y}_i = g(z_i; \phi)$$

Eqn (32)

where φ denotes the meta-parameters. The meta-learner is optimised with AdamW at a learning rate of 1×10^{-3}, weight decay 1×10^{-3} and a cosine-annealing schedule, with a budget of 85 epochs and a batch size of 32. The objective is:

$$\min_{\phi} \sum_{i=1}^{N} \mathcal{L}_{CE}\left(y_i, g(z_i)\right)$$

Because the meta-learner operates on a 32-dimensional vector rather than on image data, its training and inference costs are negligible relative to those of the four base learners, which dominate the computational budget of the framework.

In the experiments reported here, the meta-training matrix is assembled from the held-out predictions of the first cross-validation fold only, giving one 32-dimensional meta-feature vector per sample in that partition. The meta-learner is then fitted and evaluated on this same set of vectors. The resulting ensemble figures are therefore in-sample estimates and are not directly comparable with the held-out base-learner figures reported in Section 5.1; this is addressed in Section 5.4.

4.10.4 Results

Model evaluation employs standard metrics:

Accuracy: The accuracy of classification work is calculated through the equation which shows what percentage of correct predictions the model made from its total predictions. The expression defines TP as True Positives which counts all correctly predicted positive samples while TN identifies True Negatives as the correct negative predictions and FP shows the negative samples that were wrongly identified as positive and FN defines False Negatives as the positive samples which were wrongly identified as negative.

The numerator (TP+TN) counts the total number of correct predictions, while the denominator (TP+TN+FP+FN) represents the total number of evaluated samples. Therefore, accuracy computes the ratio of correct predictions to all predictions.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Eqn (34)}$$

Precision: The Precision metric of Equation 35 calculates its value through two different ı

The measurement counts all positive predictions made by the model to calculate the proportion of correct positive predictions. In this measurement True Positives (TP) counts all positive cases that have been correctly identified while False Positives (FP) counts all negative cases that have been wrongly identified as positive cases.

The denominator (TP+FP) represents all samples the model predicted as positive, while TP counts only the correct positive predictions. Precision provides assessment of positive prediction accuracy by showing what percentage of predicted positive cases were actually right.

$$Prec = \frac{TP}{TP + FP} \quad \text{Eqn (35)}$$

Recall: Equation (36) defines the Recall, which is also called the True Positive Rate (TPR) or Sensitivity. This metric evaluates how well a classification model identifies actual positive samples. The formula uses TP (True Positives) to show how many positive cases were correctly identified and FN (False Negatives) to show how many positive cases the system mistakenly identified as negative. The denominator (TP+FN) corresponds to the total number of actual positive samples, while the numerator TP counts only those that are correctly detected. The system measures its ability to find all important positive instances through its recall measurement.

$$Rec = \frac{TP}{TP + FN} \quad \text{Eqn (36)}$$

F1-Score: The F1-score which Equation (37) defines calculates classification performance through its balanced assessment of Precision and Recall metrics. The expression defines Precision as the measure which determines how many predicted positive samples are correct while Recall tracks the number of actual positive samples which systems successfully detect. The product (Precision · Recall) shows their combined performance through its calculation which gives both metrics equal weight through a multiplication of 2. The score gets normalized through the process of dividing by their total value.

The use of harmonic mean provides better results than arithmetic mean because it produces stronger penalties against extreme data points. The F1-score drops when either precision or recall falls below acceptable levels. The system achieves its goal by ensuring that both false positives and false negatives reach their lowest possible levels.

Eqn (37)

$$F1 = \frac{2(Prec \cdot Rec)}{Prec + Rec}$$

Fig. 7 Representative samples drawn from the eight coral classes.

Fig. 8 Class distribution of the curated coral dataset.

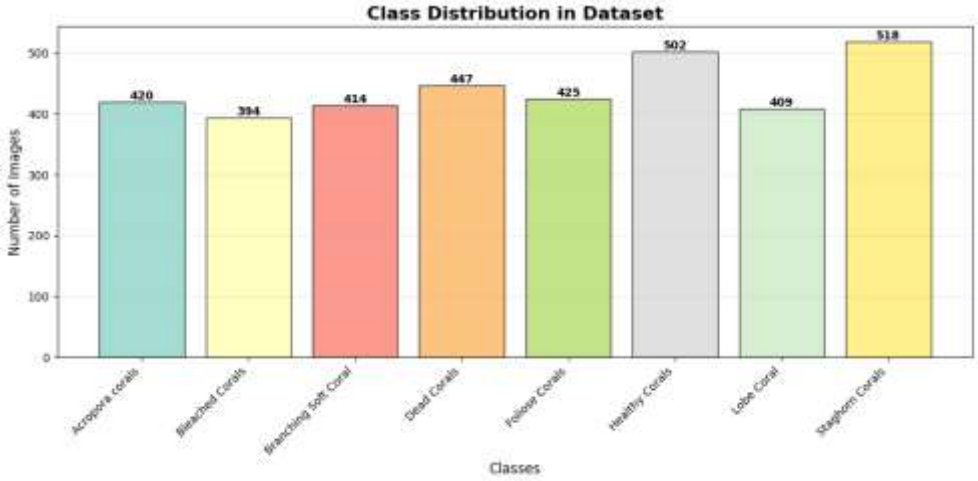


Fig. 9 Training and validation curves for the METASTACK-NET meta-learner.

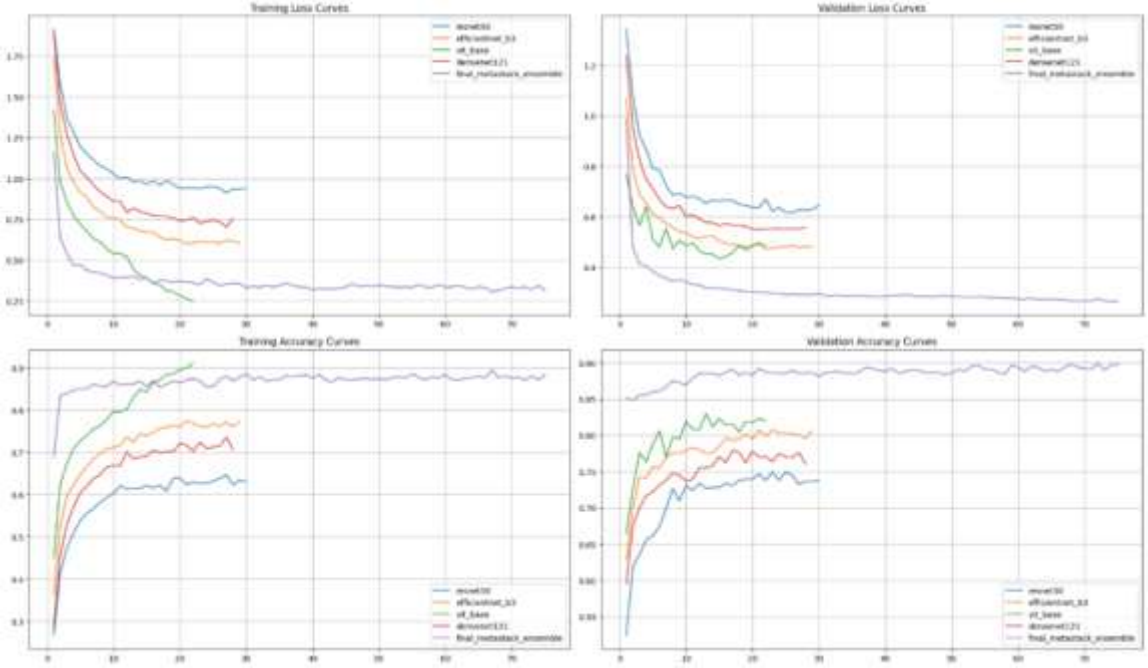


Fig. 10 Confusion matrix of METASTACK-NET on the held-out test set.

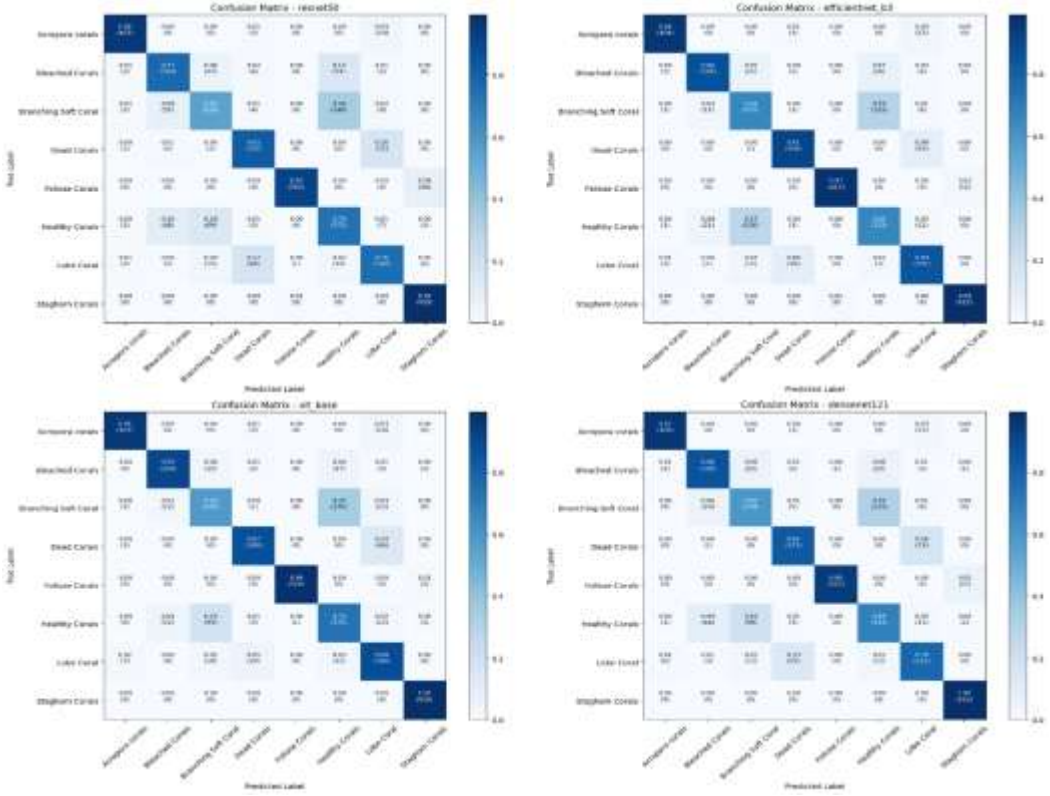


Fig. 11 Per-class ROC curves with micro- and macro-averaged AUC for METASTACK-NET.

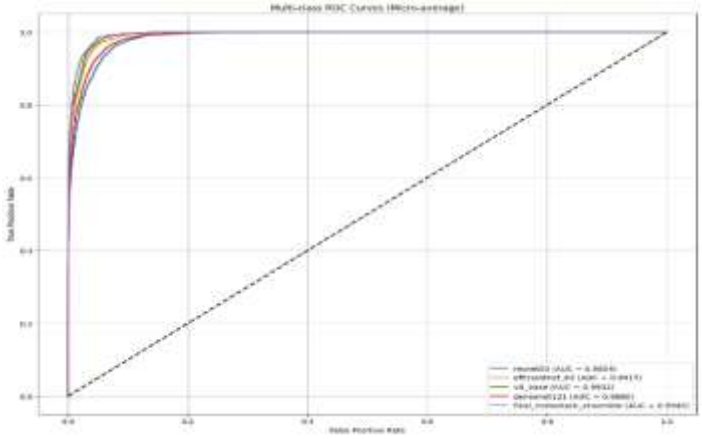
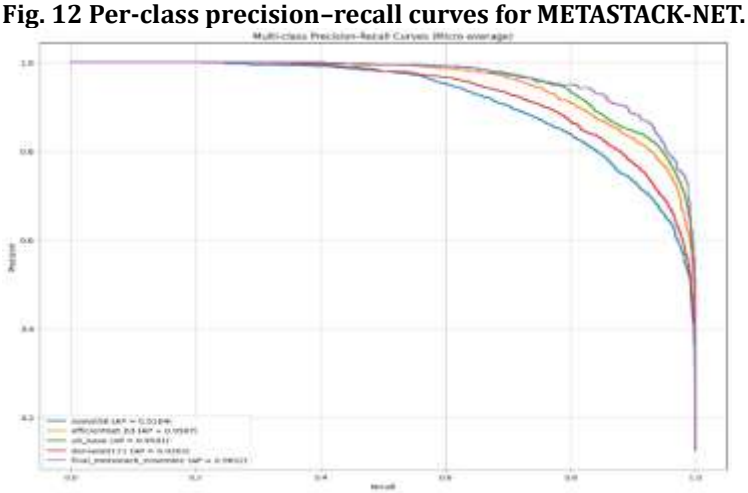


Fig. 12 Per-class precision–recall curves for METASTACK-NET.



5 Comparative Analysis and Discussion

5.1 Cross-Architecture Comparison

Table 3 Comparative performance of four modern architectures under stratified 13-fold cross-validation with out-of-fold aggregation. † The stacking ensemble figure is an in-sample estimate (Section 5.4) and is not comparable with the cross-validated rows above.

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
ResNet50	0.8284	0.8294	0.8254	0.8266	0.9779
EfficientNet-B3	0.8700	0.8705	0.8700	0.8698	0.9847
ViT-Base	0.8802	0.8800	0.8795	0.8794	0.9856
DenseNet121	0.8505	0.8499	0.8484	0.8488	0.9807
Stacking ensemble (preliminary)	**0.9128**	**0.9218**	**0.9100**	**0.9105**	**0.9834**

5.2 Per-Class Behaviour and Error Complementarity

Across both protocols the four architectures encode measurably different views of the same imagery. ResNet50 supplies robust residual features; EfficientNet-B3 achieves comparable accuracy at markedly lower parameter cost through compound scaling; DenseNet121 improves gradient propagation and feature reuse through dense connectivity; and ViT-Base captures long-range contextual dependencies through self-attention. Under identical data and protocol, ViT-Base is the strongest single architecture, which is consistent with the observation that reef scenes carry substantial global context — colony extent, substrate type, neighbouring structure — that convolutional receptive fields capture only indirectly.

The more informative result lies beneath the aggregate figures. Every architecture in both protocols recovers the morphologically distinctive classes — Acropora, foliose and staghorn — at F1-scores above 0.94, and every architecture degrades on the same three classes: branching soft, bleached and lobe coral. This is a property of the visual problem, not of any particular network: bleached and dead coral are spectrally adjacent under attenuated illumination, and branching soft coral shares textural statistics with several hard-coral morphologies. Aggregate accuracy therefore overstates operational readiness, because the classes that drive bleaching surveillance are precisely the ones the models handle worst. The failures are nonetheless architecture-specific in their distribution: on branching soft coral, recall ranges from 0.84 to 0.13 across the seven backbones of the fixed-partition evaluation, and comparable spreads appear on healthy coral. Errors are therefore only partially correlated, which is the condition under which multi-model approaches can be expected to help.

5.3 Implications for Ensemble Approaches

The comparative results establish the precondition on which ensemble methods depend. Ensembling yields no benefit when constituent models fail on the same inputs, since combining correlated errors reproduces them; it yields substantial benefit when failures are distributed differently across models, because a fusion rule can then learn which model to trust in which region of the class space. The per-class evidence in Section 5.2 shows the latter condition holds here, and holds most strongly on exactly the classes where single-model performance is weakest. Quantifying the achievable gain requires a stacking model trained on out-of-fold predictions pooled across all folds and evaluated on a partition disjoint from its training set; the formulation in Section 4.10.3 provides the starting point, and this is identified as the primary direction for subsequent work.

5.4 Limitations

Four limitations qualify this study. First, the two groups of architectures were trained under different partitioning schemes — a fixed train/validation/test split for the seven transfer-learning backbones and stratified 13-fold cross-validation for the four modern architectures — so results are compared within each scheme and are not merged into a single ranking; the cross-protocol discussion is confined to per-class

error patterns, which are not partition-dependent. Second, the exploratory stacking model of Section 4.10.3 is fitted and evaluated on the same set of meta-feature vectors, drawn from a single cross-validation fold, and its figures are therefore in-sample; they are reported for completeness and are excluded from the comparative claims of this paper. Third, the dataset, while balanced across eight classes, is drawn from a limited set of reef sites and does not span the range of seasonal, geographic and turbidity conditions encountered in operational surveys, so absolute accuracies should be read as protocol-relative rather than as expected field performance. Fourth, classification is performed on whole images rather than on segmented colonies, so scenes containing multiple coral morphologies receive a single label; extending the comparison to instance segmentation is a necessary step toward quantitative benthic cover estimation.

6 Conclusion and Future Work

This paper reported a controlled comparative evaluation of ten convolutional and transformer architectures for underwater coral reef classification on a single curated eight-class dataset. Rather than proposing a new model, the study held data, class definitions, augmentation and evaluation metrics constant so that observed differences could be attributed to architecture rather than to experimental setup — a control that is largely absent from the existing literature, where results are reported on incompatible datasets and protocols.

Three findings follow. Vision transformer representations are competitive with, and under the cross-validated protocol stronger than, the convolutional architectures tested, which is consistent with the importance of global scene context in reef imagery. Aggregate accuracy is nonetheless a weak indicator of operational readiness: every architecture recovers morphologically distinctive classes almost perfectly while failing on branching soft, bleached and lobe coral — the classes on which bleaching surveillance most depends. And those failures are only partially correlated across architectures, so the residual errors are architecture-specific rather than intrinsic to the imagery, which establishes both the opportunity for ensemble methods and the specific classes on which such methods should be evaluated.

Several directions follow directly. The most immediate is a properly validated ensemble over the architectures compared here, trained on out-of-fold predictions pooled across all folds and evaluated on a disjoint partition, using the formulation of Section 4.10.3 as its starting point. Re-evaluating all ten architectures under a single partitioning scheme would allow them to be ranked together rather than within groups. Because the enhancement pipeline described in Section 3.3 was not active during these experiments, a paired evaluation with and without underwater enhancement would establish whether image restoration improves downstream classification or only the image-quality proxies conventionally reported. Beyond this, self-supervised and contrastive pretraining would reduce dependence on scarce annotated reef imagery; model compression is required for deployment on power-limited underwater platforms; and broadening the dataset across additional species, reef systems and seasonal conditions is necessary to establish external validity.

References

[1] Akkaynak, D., Treibitz, T.: A revised underwater image formation model. Proc. IEEE CVPR, 6723–6732 (2018).

[2] Li, C., Guo, J., Guo, C., Cong, R., Liu, J.: Underwater image enhancement by dehazing with minimum information loss and histogram distribution prior. IEEE Trans. Image Process. 25(12), 5664–5677 (2016).

[3] Anwar, S., Li, C., Porikli, F.: Deep underwater image enhancement. Pattern Recogn. 98, 107038 (2020).

[4] Islam, M.J., Xia, Y., Sattar, J.: Fast underwater image enhancement for improved visual perception. IEEE Robot. Autom. Lett. 5(2), 3227–3234 (2020).

[5] Wang, Y., Huang, T.W., Li, C., Chen, J., Sun, W., Xu, L., Jin, L.: Deep learning for underwater image enhancement: A survey. IEEE Access 7, 142713–142726 (2019).

[6] Fabbri, C., Islam, M.J., Sattar, J.: Enhancing underwater imagery using generative adversarial networks. Proc. IEEE ICRA, 7159–7165 (2018).

[7] Li, J., Skinner, K., Eustice, R.M., Johnson-Roberson, M.: WaterGAN: Unsupervised colour correction for underwater images. IEEE Robot. Autom. Lett. 3(1), 387–394 (2018).

[8] Chen, J., Zhao, P., Ma, Y., Yang, J.: UWCNN: Underwater image enhancement using adaptive dense feature fusion. IEEE Trans. Circuits Syst. Video Technol. 30(12), 4709–4723 (2020).

[9] Carlevaris-Bianco, N., Mohan, A., Eustice, R.M.: Initial results in underwater single image dehazing. IEEE/MTS OCEANS, 1–10 (2010).

[10] Drews Jr., P.L., Nascimento, E.R., Moraes, F., Botelho, S.S., Campos, M.F.: Transmission estimation in underwater images. Proc. IEEE CVPR Workshops, 825–830 (2013).

[11] Jiang, W., Gu, Z., Zhao, J., Wang, H.: A comprehensive survey on underwater image enhancement. Inf. Fusion 79, 46–73 (2022).

[12] Arredondo, J.E., Vasquez, D.A., Mejia, J.F., Branch, J.W.: Underwater image enhancement using CycleGAN. IEEE LARS, 1–6 (2019).
[13] Fu, X., Zhuang, P., Huang, Y., Liao, Y., Ding, X., Paisley, J.: A Retinex-based enhancing approach for single underwater image. Proc. IEEE ICIP, 1389–1393 (2014).
[14] Guo, Y., Li, H., Yu, Z., Wang, Z., Tan, R.T., Yang, M.-H.: Underwater image restoration using multi-colour space embedding. IEEE Trans. Image Process. 29, 5309–5324 (2020).
[15] Hou, X., Gao, C., Qiao, M., Shen, X.: Joint contrast enhancement and denoising for underwater images via residual learning. Neurocomputing 455, 384–398 (2021).
[16] Zhou, H., Li, S., Yu, S., Cheng, Z.: Underwater image enhancement via multi-scale dense GANs. Neural Netw. 144, 499–510 (2021).
[17] Akkaynak, D.: Optical properties of natural waters: A review. Annu. Rev. Mar. Sci. 11, 315–338 (2019).
[18] Hou, W., Hieronymi, M.: Light scattering in water and imaging implications. Appl. Opt. 56(10), 2640–2652 (2017).
[19] Peng, Y.T., Cosman, P.C.: Underwater image restoration using blurriness & absorption. IEEE Trans. Image Process. 26(4), 1579–1594 (2017).
[20] Liu, R., Fan, X., Zhu, C., Hou, J., Luo, Z., Zhang, L.: Real-world underwater image enhancement: Challenges and solutions. Int. J. Comput. Vis. 129, 2281–2307 (2021).