Multi-Class Coral Reef Object Detection

Description

Traditional coral reef monitoring methods rely heavily on manual observation, which is time-consuming, costly, and difficult to scale. These limitations make it challenging to continuously assess reef conditions across large marine areas. Meanwhile, global coral reef conditions have declined by approximately 14%, highlighting the urgency for more efficient monitoring solutions.

This project proposes an object detection–based system that utilizes underwater coral reef images to automatically detect and classify multiple coral reef species. The system aims to support biodiversity assessment and help establish conservation priorities, particularly for vulnerable and endangered coral species.

Objectives

The primary objective of this project is to design and implement an automated system capable of Classifying coral reef species accurately using deep learning–based object detection. Technically, the project focuses on optimizing feature extraction, bounding box regression, and multi-class classification performance using Faster R-CNN.

Technologies Used

- Python
- PyTorch
- OpenCV
- Faster R-CNN
- Convolutional Neural Network (CNN)
- Scikit-Learn
- NumPy
- JSON
- Flutter

System Workflow

1. Dataset Collection Datasets were collected from public platforms such as Roboflow and other open-source repositories. The datasets were selected

based on relevance, annotation quality, and suitability for multi-class coral reef detection.

2. Dataset Preprocessing The preprocessing stage consists of several important steps Annotation Conversion from segmentation to bounding box annotations, Dataset merging, Overlapping Bounding box handling, Image Resizing.

3. Dataset Augmentation Several augmentation techniques were applied, Saturation increase by 50%, Saturation decrease by 50%, Brightness reduction by 40%, Hue shift by 30° in HSV color space

4. Model Training Train a faster r-cnn model with ResNest-50 backbone and Feature Pyramid Network (FPN) using PyTorch Framework. Hyperparameters were configured based on prior studies. The main tuned parameter was (lambda). Model was trained for 50 epoch and reached an map50 of 82%.

5. Graphical User Interface (GUI) Development A GUI was developed to simplify result visualization and improve user interaction, the GUI consist of Home Page as the main entry point, Image upload feature for coral reef images, Result Page to display detection outputs, Visualization of detected bounding boxes and class labels, Additional information display (Coral species description, Habitat information, IUCN Red List conservation status), and Navigation feature to return to previous pages

Key Features

- Detection and classification of one or more coral reef species in a single image
- Visualization of bounding boxes and class labels
- Detailed species descriptions
- Habitat information
- IUCN Red List conservation status

Challenges & Problems (Key Section)

- Limited availability of coral reef datasets
- Inconsistent dataset quality
- Difficulty distinguishing coral species from general coral reef structures
- Imbalanced class distribution
- High training cost and long training time
- Overfitting issues
- Low model performance (mAP, precision, and recall)

Solutions & Technical Decisions

- Collecting additional datasets and joining specialized forums to obtain rare coral data
- Manually reviewing and refining bounding box annotations to improve data quality
- Conducting literature reviews (journals and research papers) to ensure correct species classification
- Balancing datasets by limiting dominant classes and replacing underrepresented ones
- Investing in higher computational resources to reduce training time
- Performing extensive trial-and-error hyperparameter tuning
- Iteratively experimenting with dataset configurations and augmentation strategies

Results & Evaluation

The implemented solutions significantly improved model performance and dataset quality. The system successfully detects and classifies multiple coral reef species. However, training costs remain a major limitation due to high computational requirements.

Lessons Learned

- The critical importance of large-scale and high-quality datasets
- The value of building an intuitive GUI for AI systems
- How to distinguish between good and bad training data
- The impact of hardware limitations in real-world AI development
- Effective augmentation strategies for underwater imagery
- Practical techniques to improve model accuracy and generalization