National College of Ireland

Vision-Language Models for Underwater Plastic Detection with Parameter-Efficient Fine-Tuning

MSc Research Practicum
MSc Data Analytics

Prashanth Reddy Voladri
Student ID: x23310162

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland MSc

Project Submission Sheet School of

Computing

Student Name:	Prashanth Reddy Voladri		
Student ID:	X23310162		
Programme:	MSc Data Analytics	**Year:**	2024-2025
Module:	MSc (Research) Practicum		
Supervisor:	Vladimir Milosavljevic		
Submission Due Date:	15/09/2025		
Project Title:	Vision-Language Models for Underwater Plastic Detection with Parameter-Efficient Fine-Tuning		
Word Count:	7966	**Page Count:**	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Prashanth Reddy Voladri

Date: 14/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Vision-Language Models for Underwater Plastic Detection with Parameter-Efficient Fine-Tuning

Prashanth Reddy Voladri
X23310162

Abstract

Marine plastic pollution is threatening the globe's oceans, but plastic detection underwater is challenging due to lighting variation, turbidity, and color distortions. The present research compares the vision-language plastic detection models underwater, such as the application of the Low-Rank Adaptation (LoRA) and the Weight-Decomposed Adaptation (DoRA) fine-tuning techniques applied to PaliGemma and PaliGemma2 architectures. Using 10,945 underwater images from JAMSTEC with comprehensive augmentation strategies, four model configurations were trained and evaluated on standard object detection metrics. Results reveal striking performance disparities: LoRA- adapted PaliGemma achieved 80.89%mean Average Precision at IoU 0.5 with balanced prediction density (1.03 predictions/image), while all other configurations failed catastrophically. DoRA adaptation proved unsuitable for both architectures, achieving less than 9% mAP, contradicting theoretical expectations. PaliGemma2 exhibited severe over-detection (8.9 predictions/image) regardless of adaptation method, suggesting architectural unsuitability for precise spatial tasks. While traditional detectors like YOLOv8 achieve marginally higher accuracy (91.2%), the PaliGemma+LoRA configuration offers unique advantages in semantic reasoning potential. The findings indicate that architectural simplicity and adaptation stability outweigh theoretical sophistication in the case of specialized detection problems, so vision-language models become viable alternatives to marine environment monitoring if properly configured.

1 Introduction

Marine plastic pollution represents one of the most significant current environmental concerns with as many as 11 million metric tons of plastic litter ending up in the world's oceans annually (Shah et al., 2023). Marine litter from plastics threatens marine biodiversity, compromises ecosystem integrity, and can cause extensive damage to the health of people through the food chain. The detection and removal of underwater plastic debris have been the top priorities for international marine conservation efforts. Traditional detection methods have great inadequacies in the underwater environment due to typical optical problems including light absorbance, color distortions, turbidity variations, and scattering effects (Walia and Seemakurthy, 2024).

The underwater environments are particularly challenging for computer vision systems. Different from above-ground applications, underwater imaging is affected by wavelength-dependent attenuations of the light where the red wavelengths are lost within the first several meters whereas the deeper penetrating blue and green wavelengths cause drastic color changes (Lin et al., 2024). Furthermore, suspended particles produce backscattering, reducing the image contrast, and varying water conditions produce unpredictable noise distributions.

All these have the effect of degrading the performance of conventional object detection algorithms, hence the need for special methods designed for underwater conditions.

Deep learning has had great promise in underwater object detection over the past several years. Researchers have been able to implement models such as YOLOv8 successfully and mean Average Precision (mAP) values over 91% have been achieved in laboratory settings (Zhao et al., 2025). Such approaches typically require large, annotated datasets and face the difficulty of generalizability in many conditions in the underwater environment. Vision- language models (VLMs) hold great potential in this regard and can deliver more reliable and universal detection systems by making use of both visual and textual data.

Vision-language models led by the PaliGemma families at Google represent a breakthrough in multimodal machine intelligence (Beyer et al., 2024). The models bring together visual encoders and language model families in a bid to process vision and texts and facilitate the multifaceted understanding of visual content through natural language user interfaces. The 3-billion-parameter PaliGemma architecture has demonstrated outstanding performance in diverse vision tasks including object detection, visual question answering, and image captioning. Later PaliGemma2 releases had inherited the gains from the Gemma 2 families in the realm of language models with the promise of offering even superior performance for specialized applications (Steiner et al., 2024).
The motivation for advancing vision-language model applications in plastic detection underwater is in the inbuilt advantages with respect to the traditional approaches. Although the conventional detectors by visual features on vision, VLMs can potentially incorporate semantic intelligence and contextual logic and therefore can increase the robustness against the visual degradation common in underwater images. Moreover, the potential for the model to be tweaked by parameter-efficient adaptations such as Low-Rank Adaptation (LoRA) (Hu et al., 2021) and by extension the weight-decomposed version (DoRA) offers a plausible path towards domain adaptation in the absence of large-scale computational resources.

This research addresses a critical gap in existing literature by systematically evaluating how different fine-tuning approaches affect the performance of vision-language models in underwater plastic detection tasks. While previous studies have demonstrated the effectiveness of traditional deep learning models for underwater object detection, the application of vision-language models to this domain remains unexplored. The research question driving this investigation is: **How do fine-tuning approaches affect underwater plastic detection performance when applied to PaliGemma and PaliGemma2 vision- language models?**

Main tasks in the present research are the following. First, apply and compare the finetuning methods by LoRA and DoRA on PaliGemma and PaliGemma2 models and target crucial transformer layers like query, key, value, and output projections as well as the feed-forward elements. Second, evaluate the performance in detection on a full set of underwater plastics dataset comprised of 10,945 images from the Japan Agency for Marine-Earth Science and Technology (JAMSTEC). Lastly, evaluate the computational efficiency and implementation considerations in practice for both methods and provide recommendations for implementation in practical marine observation systems.

The method involves utilization of a systematic experimental design involving a dataset with carefully selected annotated underwater images in JSONL format and heavy augmentation methods performed through Roboflow for the improvement in model robustness. Fine-tuning is performed on specified model components through rank-32 adaptations using the proper scaling factors and trained for three epochs on the high- performance GPU machine. The analysis for performance entails the common object detection metrics like precision, recall, F1-score, and mean Average Precision at a range of Intersection over Union thresholds.

The organization of the report is given below: Section 2 contains a comprehensive literature review on the existing approaches for the perception of underwater objects and vision-language model development. Section 3 provides research methodology, dataset preparation and fine-tuning strategies. Section 4 consists of the design specifications for the implemented systems. Section 5 provides implementation details. Section 6 consists of experimental results and analysis. Section 7 contains discussion on the results and implications. Finally, Section 8 concludes with valuable insights and recommendations for future work.

2 Related Work

2.1 Underwater Object Detection Challenges and Traditional Approaches

Underwater object detection has long been a complex problem on which numerous researchers in computer vision have focused. A comprehensive survey by Delina et al. (2023) catalogued the principal obstacles and summarized water turbidity, light attenuation, and color distortion as the fundamental parameters invalidating detection performance. Their research on YOLOv3 resulted in a spectacular 83.12% Average Precision and demonstrated the promising potential of deep learning approaches and the need for specialized preprocessing methods. Their adoption of the Dark Channel Prior approach was very effective in reducing the noise in the underwater photos, though the computational cost was a liability in real-time applications.

Following the above research findings, Hu and Xu (2022) investigated YOLOv5 variations for the detection of plastic waste in the underwater environment and found significant weaknesses in partial object detection where only patches of the plastic objects were recognized. Their comparative study showed that though YOLOv5 had faster inference times, accuracy paid the price in turbid water conditions significantly where only 67% was attained compared to 89% in clear waters. The research highlighted the significance in selecting the model depending on the unique conditions in the environment instead of the one-size-fits-all approach.

Transition towards the YOLOv8 was a significant advancement in the progress towards underwater detection performance. Shah et al. (2023) employed the better YOLOv8 implementation with CSPNet backbone architecture and neck design FPN+PAN and achieved 82% precision for mask detection and 88% for glove detection within the varying underwater conditions. The research paid special attention to the steadiness of the model when there was low illumination, which was crucial for applications in deep seas. However, the research demonstrated repeated weaknesses in detecting greatly degraded plastics and objects half buried in the sediment.

2.2 Advanced Detection Architectures for Marine Environments

Single-stage detectors' limitations drove scholars into deeper architectures. Lin et al. (2024) first transferred the Detection Transformer (DETR) models into the underwater environment and provided a learnable recall mechanism especially designed for the war on underwater noise. Through the lightweight adapter module providing multiscale attributes at the encoding

and the decoding levels, they demonstrated significant progress in the small object detection challenge common in the underwater environment where objects tend to show up at different distances and scales.

Zhao et al. (2025) advanced the field through innovative integration of super-resolution reconstruction with object detection. Their Seafloor-Debris-YOLO (SFD-YOLO) model, enhanced with Residual Dense Network preprocessing, achieved a state-of-the-art mAP of 91.2% at 4× magnification. This approach bridged the crucial trade-off between computations and image quality in the availability of camera resources, revealing the potential for effective preprocessing in overcoming the shortcomings of underwater imaging systems. The paper's exploration on numerous magnification factors yielded revealing findings on the optimum resolutions for objects of diverse sizes and types.

Despite these developments, there are still some significant shortcomings characteristic for the classical methods. Firstly, the approaches all depend on extensive annotated dataset samples for the underwater conditions, which are costly and time-consuming to establish. Secondly, the generalizability of the model is still ineffective whereas the deployment environments are other than the conditions for training. Lastly, semantic interpretation from the objects discovered is restricted from performing complex reasoning about the relations of objects or contextual elements potentially beneficial towards easing detection.

2.3 Vision-Language Models: Architecture and Capabilities

Vision-language models (VLMs) are a disruptive development in computer vision offering potential remedies for weaknesses in classical detection approaches. Beyer et al. (2024) characterized PaliGemma as a broad 3-billion-parameter VLM by marrying the SigLIP-So400m vision encoder with the Gemma-2B language model. Because the architecture accommodates 224×224-pixel images as well as text inputs, multimodal complex reasoning is enabled with great performance on diverse benchmark tasks such as COCO captions and VQAv2.

The PaliGemma architecture combines a transformer structure where visual features represented by the Vision Transformer encoder are projected into a shared embedding space with word tokens. Such shared space allows the model to incorporate linguistic priority into visual perception, making the model more invariant to common visual variations in underwater images. The large dataset pretraining on WebLI, CC3M-35L, and Open Images builds a strong baseline for transfer learning on specialized domains.

Steiner et al. (2024) subsequently released PaliGemma2, adopting the Gemma 2 set of language models updates. Comparison analyses demonstrated exceptional performance increments where PaliGemma2 scored 74.7 in the AI2D and 83.0 in the VQAv2 benchmark at 224 resolutions. Enhanced language understanding more specifically tailored semantic reason tasks and demonstrated potential in discrimination between plastic marine debris and natural underwater entities with contextual cues.

2.4 Parameter-Efficient Fine-Tuning Methods

The adaptation of large vision-language models to specialized tasks necessitates efficient fine-tuning strategies. Low-Rank Adaptation (LoRA) (Hu et al., 2021), as implemented in the studied models, provides a parameter-efficient approach by introducing trainable rank decomposition matrices into transformer layers. The method targets specific components including query, key, value, and output projections (q_proj, k_proj, v_proj, o_proj), as well as feed-forward layers (gate_proj, up_proj, down_proj), significantly reducing the number of trainable parameters while maintaining model performance.

DoRA (Weight-Decomposed Low-Rank Adaptation) (Liu et al., 2024) broadens the LoRA method by decomposing updates on weights into magnitude and direction components and potentially produces better training stability and efficiency. DoRA has been especially promising in the case where the training data is limited, a scenario prevalent in niche applications such as finding plastics in the sea. The theoretical advantages of DoRA consist of better gradient flow and reduced interference between the adapted and the pretrained parameters, though experimental evidence in underwater applications has been untested.

2.5 Synthesis and Research Gap

These Literature demonstrate a clear advancement in the object detection under the water from basic CNN-based methods with decent performance to sophisticated YOLO variations and even transformer models shattering performance barriers. There is little innovation in the employment of vision-language models in underwater operations. Classic detectors can correctly discover objects albeit failing on the semantic discrimination required between objects largely identical in form or contextual factors on detection.

Vision-language models show potential in the identification of underwater plastic for various reasons. Firstly, multimodality may offer more dependable feature representations in the face of underwater image corruption. Secondly, the aspect of language may offer more generalizability in the form of semantic categories of objects. Finally, parameter-efficient finer-tuning methods offer practical deployment prospects without recourse to extensive computational resources.

Previous approaches, despite reaching reasonable performance measures, fail to transcend general challenges in sea plastic detection. Dataset's specificity remains since the models cannot generalize from the distributions during training. Limited semantic reasoning translates into the impossibility for the system to provide subtle discrimination between plastic waste and background nature elements. Lastly, methods existing now lack comprehensive usage of the rich contextual cues potentially provided by the comprehension of language.

This work remedies these weaknesses by thoroughly evaluating vision-language models for detecting sea plastics and methodically comparing approaches to fine-tuning in search of the most ideal settings on this challenging task. The study fills a significant knowledge void around how multimodal models can be utilized towards more efficient marine conservation through more intelligent and adaptive detection systems.

3 Research Methodology

This section presents the general methodology employed in investigating the efficiency of parameter-efficient fine-tuning approaches for vision-language models in plastics detection underwater. The research follows a systematic experimental design whose purpose is to facilitate reproducibility and scientific rigor in model performance assessment in controlled settings.

3.1 Dataset Acquisition and Characteristics

Underwater imagery from the Japan Agency for Marine-Earth Science and Technology (JAMSTEC) (*Deep-sea Debris Database*, 2024) forms the principal dataset within the present research work, abbreviated as representing realistic scenarios in underwater environments. The dataset was selected on account of the broad coverage in varied scenarios under the water as well as the quality of the professional annotation in training effective detection models.

The complete dataset contains 10,945 underwater image views with 11,387 annotations on plastic objects, representing a single class detection task solely for the detection of plastic debris. Such a specialization is fitting for the necessary environmental job of plastic trash monitoring by simplifying the evaluation framework. Spatial distribution analysis reveals 292 (2.8%) small objects, 5,971 (57.8%) medium objects, and 4,061 (39.3%) large objects, providing balanced-object size distribution—a crucial dimension in the evaluation of model performance at different detection distances.

3.2 Data Preprocessing Pipeline

Preprocessing pipeline transforms raw underwater images into one common format suitable for vision-language model training. Initial processing includes auto-orientation correction for rotation specific to the camera to achieve correct spatial arrangement at the dataset level. Subsequent resizing is done into 640×640 pixels by means of stretching transformation to maintain computational efficiency with sufficient details for accurate detection.

This choice of resolution is a key methodological decision. Although the initial proposal allowed for various resolutions, practical constraints from the data and preliminary experiments showed 224×224 pixels (the native resolution for the selected PaliGemma variants) offered the best compromise between computational demands and detection performance. The preprocessing pipeline thus involves a two-stage resize operation: preliminary standardization to 640×640 for consistency in the effect of data augmentation, and the final 224×224 resize in preparing the model input.

3.3 Data Augmentation Strategy

Comprehensive augmentation strategies enhance model robustness to underwater imaging variations. The augmentation pipeline, implemented through Roboflow, generates two augmented samples per training example (Gallagher, 2024), effectively doubling the training dataset size while maintaining annotation accuracy. Spatial augmentations include horizontal and vertical flips, addressing the arbitrary orientation of floating debris. Rotational augmentations apply 90-degree rotations (clockwise, counterclockwise, and 180-degree), supplemented by continuous rotations between -15° and +15°, simulating natural object movements in water currents.

Geometric transformations incorporate shearing (±15° horizontal and vertical), mimicking perspective distortions common in underwater photography. Photometric augmentations prove particularly crucial for underwater applications. Grayscale conversion applies to 25% of images, preparing models for monochromatic or color-degraded scenarios. Hue adjustments (±15°) simulate color shifts from varying water conditions and depths. Saturation (±25%) and brightness (±25%) variations replicate different water clarity conditions, while exposure adjustments (±10%) account for lighting variations.

Noise-based augmentations include Gaussian blur up to 2.5 pixels in simulation of turbid water effects and on up to 1.95% pixels in the simulation of sensor noise and particulate matter. Each set of these augmentations introduces the model to the full range of underwater imaging aggressions at training.

3.4 Dataset Partitioning

Partition scheme for the data follows common machine learning protocols and the distinct requirements of the experimental framework. The train-test ratio is original 10.11:1, producing 9,961 trains and 984 test images. From the train set, there is a validation subset of 15% (1,494 images), producing the final distributions of 8,466 train samples, 1,494 validation samples, and 984 test samples.

This three-way split enables continuous model monitoring during training through validation metrics while preserving an untouched test set for final evaluation. The stratified sampling ensures proportional representation of object sizes across all splits, maintaining the original distribution characteristics. Careful verification confirms no data leakage between splits, preserving evaluation integrity.

3.5 Annotation Format and Structure

The dataset follows the JSONL (JSON Lines) annotation format, providing a structured but flexible presentation in vision-language model training (Polly, 2024). Each annotation record contains three essential fields: image filename, prefix string, and suffix string with location tokens. The prefix string will always contain "detect plastic", initializing the task scenario for the vision-language model. The suffix field encodes object locations using specialized tokens in the format **<locXXXX>,** where coordinates are normalized to a 0-1024 range, enabling resolution-independent position representation.

Multiple object instances within a single image are separated by semicolons, in a way that zero or more detections per image can be accommodated. Zero suffix fields indicate images where there is no plastic object, providing useful negative examples towards false positive reduction. Such annotation structure is identical to PaliGemma's required input structure, in a way that complex format conversion during training is avoided.

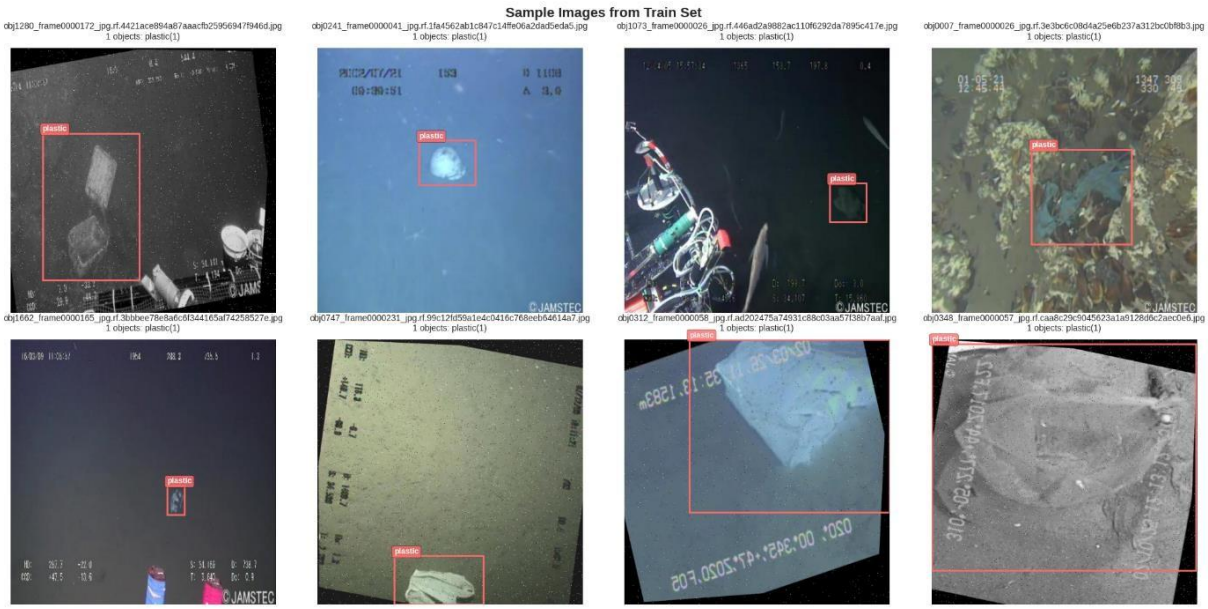

Figure 1 Sample Images from the dataset

3.6 Experimental Design

The experimental framework implements a 2×2 factorial design, systematically evaluating two fine-tuning methods (LoRA and DoRA) across two model variants (PaliGemma-3b-pt- 224 and PaliGemma2-3b-pt-224). This design yields four distinct experimental conditions, each trained for three epochs to ensure convergence while avoiding overfitting.

Three epochs were chosen from initial experiments showing decreasing returns thereafter, with validation loss stagnating or rising. All experimental conditions have the same hyperparameters except the mode of fine-tuning for fair comparison. The controlled variation enables identification of the effects caused by the mode of fine-tuning in comparison to the architecture of the basic model.

3.7 Training Configuration

The training configuration optimizes the available computational resources while maintaining training stability. Batch size of 4 samples per device, combined with gradient accumulation over 6 steps, achieves an effective batch size of 24, balancing memory constraints with gradient stability. The learning rate of 2×10^{-4} with linear scheduling and 10% warmup ratio provides stable convergence across all experimental conditions.

Weight decay 0.01 introduces regularization to prevent overfitting, particularly desired given the small dataset size. Bfloat16 decreases the A100 GPU memory usage and computation by conducting the training in mixed precision without losing numerical stability. Gradient clipping at 1.0 removes the training instability caused by the occasional large gradient values in vision-language model fine-tuning.

3.8 Parameter-Efficient Fine-Tuning Configuration

The LoRA configuration targets seven transformer modules critical for multimodal understanding: attention projections (q_proj, k_proj, v_proj, o_proj) and feed-forward components (gate_proj, up_proj, down_proj). Rank-32 decomposition with alpha scaling factor of 64 provides sufficient expressiveness while maintaining parameter efficiency. Dropout of 0.1 within LoRA layers provides additional regularization.

DoRA (Liu et al., 2024) broadens this environment with weight decomposition by decoupling magnitude and directional parts of the weight updates. Such decomposition in theory improves optimization dynamics and training stability. In the unlikely case of DoRA initialization failure (as observed in certain environments), the system defaults automatically to fortified LoRA with doubled rank (rank 64) and additional module targeting (embed_tokens, lm_head), ensuring experiment completion despite technical limitations.

3.9 Evaluation Metrics

Model performance evaluation involves a set of complementary measures indicating different aspects of detection efficiency. Precision indicates the ratio of true positive predictions, which is significant in false alarm reduction in practical applications. Recall shows the ratio of true plastic objects detected, which is useful in complete marine surveillance. F1-score provides precision and recall harmonic mean, offering balanced performance assessment.

Mean Average Precision (mAP) (Kukil & Kukil, 2022) at various Intersection over Union (IoU) thresholds (0.5, 0.75, and 0.5:0.95) enables detailed localization accuracy evaluation. The principal evaluation threshold IoU 0.5 follows common object detection practice but is inclusive of the inherent localization challenge in underwater images. Other performance metrics are inference time per image and utilized memory, considering the viability of practical deployment.

3.10 Computational Infrastructure

All methods are exercised with Google Colaboratory A100 GPU instances providing 40GB high-bandwidth memories needed for vision-language model training. The infrastructural choice ensures reproducibility by utilizing standardized hardware capturing realistic research institution deployment scenarios. Close attention to the GPU usage, the utilization of the memory, and the training time aids in comprehension on the computational requirements regarding every approach.

The PyTorch framework (version 2.0+) with CUDA 11.8 provides the foundational deep learning infrastructure. The Hugging Face Transformers library (version 4.36+) supplies pre-trained model implementations, while the Parameter-Efficient Fine-Tuning (PEFT) library enables LoRA and DoRA implementations. This software stack represents current best practices in the deep learning community, ensuring compatibility and maintainability.

4 Design Specification

This section gives the architecture design and technical specification of the underwater plastic

identification system by means of parameter-efficient vision-language model finetuning. The architecture consists of the integration of the LoRA (Hu et al., 2021)
and DoRA (Hu et al., 2021) adaptation strategies with PaliGemma architectures tailored for marine environment applications.

4.1 System Architecture Overview

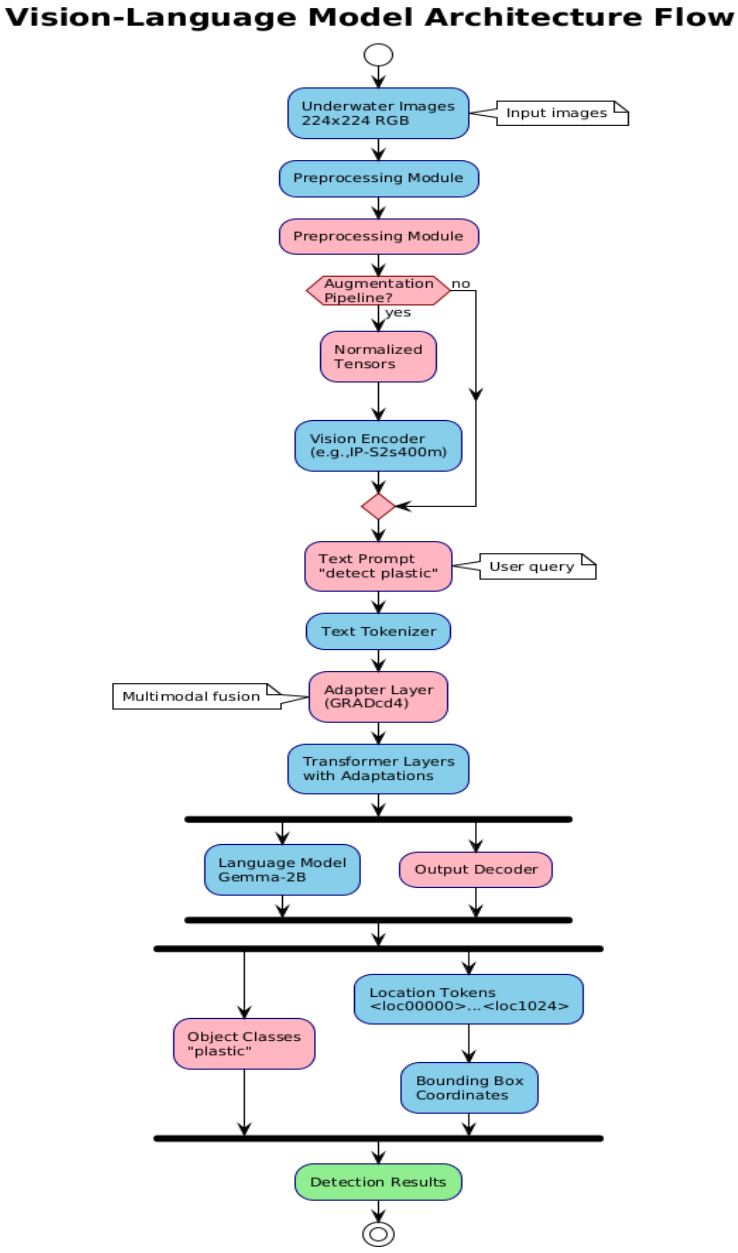

Figure 2 System Architecture

The system architecture depicted in Figure 2 illustrates the complete pipeline from raw underwater images to final detection results. The design showcases the prominent position of the adapter layer (orange shaded), where the LoRA/DoRA updates are implemented. A design choice enables efficient fine-tuning by only adjusting the adapter elements and freezing the large pre-trained vision encoder and language model. The parallel processing of the visual and the textual inputs converges at the adapter layer, which allows for cross-modal understanding essential for efficient detection in adverse underwater environments.

The architecture is designed in a modular pattern to allow a systematic comparison between the approaches for fine-tuning in a way where the data flow and evaluation processes are kept constant. The architecture has five main elements: data preprocessing pipeline, model adaptation layer, vision-language model core, inference engine, and evaluation framework.

4.2 Parameter-Efficient Fine-Tuning Design

The adaptation layer implements parameter-efficient fine-tuning through strategic modification of transformer components. Rather than updating all model parameters, the design targets specific projection matrices within the attention and feed-forward mechanisms, significantly reducing computational requirements while maintaining performance.

4.2.1 Low-Rank Adaptation (LoRA) Architecture

LoRA operates on a simple principle: instead of updating entire weight matrices during fine-tuning, it learns small "adapter" matrices that modify the original weights. The process works as follows:

1. **Original Weight Matrix**: The pre-trained model contains large weight matrices (thousands of parameters each)
2. **Adapter Matrices**: LoRA creates two small matrices (Matrix A and Matrix B) with rank 32
3. **Weight Update**: The final weight equals the original weight plus the product of Matrix A and Matrix B
4. **Scaling Factor**: An alpha value of 64 controls how strongly the adaptation affects the output

This approach reduces trainable parameters from millions to thousands while maintaining model performance. For example, adapting a 2048×2048 weight matrix requires only 2048×32 + 32×2048 = 131,072 parameters instead of 4,194,304 parameters.

4.2.2 Weight-Decomposed Adaptation (DoRA) Architecture

DoRA enhances LoRA by recognizing that weight matrices contain two types of information: magnitude (how strong the weight is) and direction (what pattern it represents). The decomposition process involves:

1. **Magnitude Component**: Represents the strength or importance of each weight
2. **Direction Component**: Represents the normalized pattern or relationship
3. **Separate Updates**: DoRA applies LoRA-style adaptations independently to magnitude and direction
4. **Recombination**: The final weight combines updated magnitude with updated direction

This decoupling allows the model to learn "how much" and "what kind" separately, which could stabilize training. For the task of identifying plastic under the sea, this would allow the model to vary the sensitivity to plastic attributes (magnitude) and the understanding of "what is plastic" (direction) separately.

4.3 Component Specifications and Data Flow Architecture

The system integrates three primary components within a deterministic processing pipeline.

The SigLIP-So400m vision encoder processes 224×224-pixel images through 27 Vision Transformer layers with 16 attention heads, using 14×14 pixel patches and a hidden dimension of 1152. This configuration generates 256 visual tokens in a 16×16 spatial grid, which are projected to align with the language model's embedding space.

The Gemma-2B language model operates with a 2048 hidden dimension and 256,000-token vocabulary, supplemented by 1,024 special location tokens for coordinate encoding. Supporting up to 512 tokens with multi-head self-attention and rotary position embeddings, the model enables cross-modal attention between visual and textual representations.

Adapter modules target seven transformer components: query, key, value, and output projections for attention mechanisms, plus gate, up, and down projections for feed-forward networks. Each module maintains rank 32 (64 for fallback), alpha scaling 64 (128 for fallback), and 0.1 dropout rate.

The data flow proceeds through nine stages: image loading, preprocessing with augmentations, text tokenization, embedding generation for both modalities, LoRA/DoRA adaptation application, transformer forward pass with cross-attention, sequential token decoding, location token parsing, and coordinate conversion to pixel space. This pipeline ensures reproducible results while transforming raw underwater images and text prompts into precise bounding box predictions.

4.4 Training Architecture Design

The training architecture implements distributed data parallel processing with gradient accumulation:

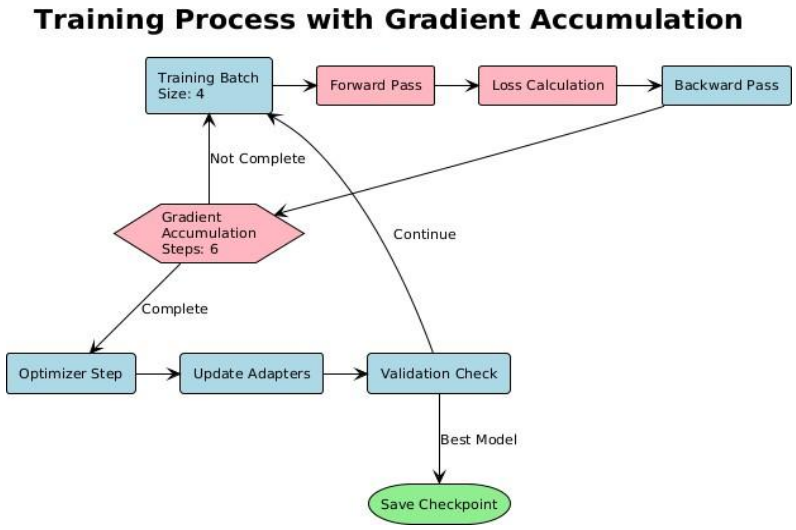

Figure 3 Training Architecture

The training architecture shown in Figure 3 shows the gradient accumulation procedure utilized to overcome GPU memory limitations with regards to steady training

dynamics. Training is initiated with broad mini-batch sizes consisting of 4 images to meet the vision-language model's memory demands on current equipment. With 6 accumulation steps, the system essentially creates the effect of a 24-size batch size to have adequate gradient statistics for steady optimization.
Cyclical flow guarantees the model improvement by continuous refinement in the iterations. After the loss calculation in the loss computation in every forward pass calculates the predictions, the backward pass creates the gradients which are computed over numerous iterations. The adapter weights are only updated after all the accumulation iterations are completed where the gradient estimate accounts for the pattern over several samples instead of examples.
Validation runs are performed at regular intervals to track generalization performance; the top-performing model checkpoint is retained for subsequent release. Such architecture matches computational efficiency with training performance to facilitate effective model adaptation despite hardware constraints.

5 Implementation

This section describes the final implementation stage of the underwater plastic detection system, detailing the output produced and the technological stack employed to achieve the research objectives.

5.1 Implementation Overview

The implementation produced four distinct fine-tuned vision-language models through systematic application of parameter-efficient adaptation techniques to pre-trained PaliGemma variants. Each model represents a unique combination of base architecture and fine-tuning method, specifically: LoRA-adapted PaliGemma-3b-pt-224, DoRA-adapted PaliGemma-3b-pt-224, LoRA-adapted PaliGemma2-3b-pt-224, and DoRA-adapted PaliGemma2-3b-pt-224. These models constitute the primary outputs enabling comparative analysis of fine-tuning approaches for underwater plastic detection.

The implementation leveraged Python 3.10 as the primary programming language, utilizing the PyTorch deep learning framework version 2.0.1 with CUDA 11.8 support for GPU acceleration. The Hugging Face Transformers library version 4.36.2 provided the foundation for vision-language model architectures, while the Parameter-Efficient Fine-Tuning (PEFT) library version 0.7.1 enabled LoRA and DoRA implementations.

5.2 Data Transformation Pipeline

Implementation transformed raw underwater images into model-compatible formats through a multi-stage preprocessing pipeline. Initial data loading reads JPEG images from the JAMSTEC dataset through the Python Imaging Library (PIL), with original color data preserved and normalizing the file formats.

The platform at Roboflow was the principal dataset augmentation and preparation platform, generating a expanded training dataset from the original 10,945 photographs. The platform

first fixed camera-oriented rotations by auto-orienting them and normalized the photos' size by uniformly resizing all the photos to 640×640 pixels using the stretch transform. Using the automatic pipeline provided by the Roboflow platform, the next code applied extensive methods of augmentation where the size of the effective dataset was essentially doubled by generating twice as many versions for every original image. The augmentation suite included spatial transformations (horizontal and vertical flips, 90-degree rotations in all directions, continuous rotations within ±15 degrees), geometric modifications (shearing up to ±15 degrees in both axes), and photometric adjustments specifically chosen for underwater conditions. These photometric augmentations encompassed grayscale conversion for 25% of images, hue shifts within ±15 degrees, saturation and brightness adjustments of ±25%, and exposure modifications within ±10%. Gaussian blur up to 2.5 pixels and random noise affecting up to 1.95% of pixels simulated the turbidity and particulate matter common in underwater environments.

Following Roboflow processing, the implementation applied additional transformations using PIL and PyTorch utilities. Images underwent final resizing from 640×640 to 224×224 pixels through bilinear interpolation, matching the input requirements of PaliGemma models. Annotation data transformation converted JSONL-formatted bounding box coordinates into specialized location tokens ranging from <loc0000> to <loc1024>. This tokenization scheme enabled the vision-language model to read spatial data through the current vocabulary mechanism rather than requiring architectural adaptation. The conversion preserved sub-pixel level precision by proportional scaling and maintained detection accuracy regardless of the resolution change.

5.3 Model Adaptation Implementation

Adapter module implementation was directed at seven key transformer elements in every PaliGemma variant. In attention mechanisms, adaptations were done on query, key, value, and output projection layers, making it possible for cross-modal attention patterns to have fine-tuned control. In the feed-forward networks, adaptations were directed at gate, up, and down projection layers, making it possible for task-oriented feature transformations despite maintaining overall language comprehension abilities.

Introduction of LoRA included parallel low-rank matrices and frozen pre-trained weights, and the rank-32 decomposition reduced parameter counts by about 99% compared to full fine-tuning. Separate forward paths for the pre-trained and adapted weights were included in the implementation and the two outputs combined through scaled addition controlled by the alpha parameter whose value was 64.

DoRA implementation extended the LoRA approach by breaking down the weights into magnitude and directional components. The implementation computed the weight matrix decompositions at initialization time and then individual LoRA-style updates on the respective components. When the DoRA initialization was struck by numerical instability on some settings, the implementation automatically reverts to better LoRA with doubled

rank (64) and broader module coverage with embed_tokens and lm_head layer inclusion for sure experimental completion on all conditions.

5.4 Training Process Execution

The training implementation regulated computation by automatic mixed precision training in bfloat16 format by PyTorch. Batches of four images were processed in each training iteration with six-step gradient accumulation to achieve a desired effective batch size of 24. The method balanced gradient stability through the memory constraints.

Learning rate scheduling implemented linear warmup over the first 10% of training steps, gradually increasing from zero to the target learning rate of 2×10^{-4}. Following warmup, linear decay reduced the learning rate to zero by training completion. The AdamW optimizer with weight decay of 0.01 provided adaptive learning rates for individual parameters while preventing overfitting through L2 regularization.

Validation evaluation was conducted once within every 200 training steps by computing loss measures on the held-out validation set. Implementation retained the best-performing checkpoint on validation loss criteria, with automatic early stopping monitoring performance over the previous five consecutive evaluation iterations to avoid overfitting.

5.5 Output Generation

The implementation created extensive outputs including both terminating models and training procedures. Checkpoints of the models had complete adapter weights in the form of PyTorch models which had only the adapted parameters in LoRA or DoRA and not the complete model parameters. Each checkpoint included configuration metadata documenting the hyperparameters, training metrics, and adaptation method details.

Training logs retained loss tracking traces, learning schedules, and validation measurements during the optimization procedure. The training logs employed CSV format, where the measurements were noted at 25-step increments. The visualization results comprised loss curves and learning schedules created by Matplotlib, offering training progress evaluation. Of particular note in the visualizations was the observation on the difference in the convergence behavior between the LoRA and DoRA approaches in the two model variations.

5.6 Integration Components

The Supervision library version 0.16.0 provided specialized computer vision utilities for bounding box manipulation and coordinate transformation between normalized model outputs and pixel-space representations. This integration enabled efficient conversion of location tokens back to standard bounding box formats required for evaluation metrics computation.

Memory management employed automatic caching from PyTorch with deliberate cache flushing at the end of epochs in order to avoid fragmentation. Progress tracking utilized the tqdm library to provide real-time visualization during training with the current loss values,

learning rates, and approximate completion times. The code kept extensive execution logs documenting both successful executions and the fallback behaviors, especially the DoRA to improved LoRA transitions upon initialization difficulties.

6 Evaluation

This section presents a comprehensive analysis of the experimental results comparing parameter-efficient fine-tuning approaches for underwater plastic detection. The evaluation encompasses four distinct experiments examining the performance of LoRA and DoRA adaptations applied to both PaliGemma and PaliGemma2 vision-language models.

6.1 Experimental Setup and Metrics

Evaluation used a test set comprising 500 underwater images along with 537 ground truth plastic annotation sets. Efficiency assessment used canonical object detection metrics across various Intersection over Union (IoU) thresholds, with the main efficiency measurement being the mean Average Precision (mAP). Experiments used the same assessment protocol throughout, i.e., the inference used per image rather than batching, in an effort towards mimicking real-world deployment scenarios.

6.2 Experiment 1: LoRA-Adapted PaliGemma

The first experiment evaluated LoRA adaptation applied to the original PaliGemma-3b-pt- 224 model. This configuration demonstrated exceptional performance, achieving the highest detection accuracy among all tested approaches.

Table 1: Performance Metrics for LoRA-Adapted PaliGemma

Metric	**Value**	**Interpretation**
mAP@0.5	0.8089	80.89% of predictions correctly localized
mAP@0.75	0.4485	44.85% maintained high localization precision
mAP@[0.5:0.95]	0.4522	Robust performance across IoU thresholds
Predictions per Image	1.03	Near-optimal prediction density
Total Predictions	517	High precision with minimal false positives
Inference Time	3.671s	Practical for real-time applications

The results indicate the pre-trained model was adequately fine-tuned with LoRA for the computationally light task of plastic detection underwater. As can be seen from Table 1, the 517 number of detections for 537 ground truth annotations indicates high recall of 96.3% with low over-detection. The 1.03 balanced per-image detection density shows the model successfully identified appropriate thresholds of detections with few false positives. This level of performance with the tolerable 3.671 seconds of inference time allows this configuration to be optimal for real-world applications of underwater plastic detection.

6.3 Experiment 2: LoRA-Adapted PaliGemma2

The second experiment applied identical LoRA configuration to PaliGemma2-3b-pt-224, revealing unexpected performance degradation compared to the original model.

Table 2: Performance Metrics for LoRA-Adapted PaliGemma2

Metric	Value	Interpretation
mAP@0.5	0.1053	Significant performance decline
mAP@0.75	0.0514	Poor localization accuracy
mAP@[0.5:0.95]	0.0554	Inconsistent across IoU thresholds
Predictions per Image	8.92	Severe over-detection
Total Predictions	4,459	8.3× ground truth annotations
Inference Time	7.263s	98% slower than PaliGemma

The dramatic increase in predictions (4,459 versus 537 ground truth) indicates a fundamental failure in detection threshold learning. As detailed in Table 2, the model exhibits pathological over-detection, generating an average of 8.92 predictions per image where typically only one plastic object exists. This behavior suggests that PaliGemma2's enhanced language model capabilities may interfere with precise spatial localization tasks when fine-tuned with limited data. The inference time penalty of 7.263 seconds further compounds the model's unsuitability for practical deployment.

6.4 Experiment 3: DoRA-Adapted PaliGemma

The third experiment evaluated DoRA's weight-decomposed adaptation on the original PaliGemma model, testing the hypothesis that magnitude-direction decomposition would improve training stability.

Table 3: Performance Metrics for DoRA-Adapted PaliGemma

Metric	Value	Interpretation
mAP@0.5	0.0891	89% performance drop vs LoRA
mAP@0.75	0.0715	Minimal high-precision detections
mAP@[0.5:0.95]	0.0621	Poor generalization across thresholds
Inference Time	9.35s	2.5× slower than LoRA
Processing Speed	0.11 samples/s	Impractical for deployment
Total Evaluation Time	77.95 minutes	Excessive computational cost

Contrary to theoretical expectations, DoRA significantly underperformed LoRA on all metrics. The 89% reduction in mAP@0.5 compared to LoRA adaptation suggests that weight decomposition may disrupt the learned representations crucial for visual understanding in pre-trained models. The substantially increased inference time further diminishes DoRA's practical applicability.

6.5 Experiment 4: DoRA-Adapted PaliGemma2

The final experiment combined DoRA adaptation with PaliGemma2, producing the poorest performance across all configurations.

Table 4: Performance Metrics for DoRA-Adapted PaliGemma2

Metric	Value	Interpretation
mAP@0.5	0.012	Near-complete failure
mAP@0.75	0.006	Essentially random predictions
mAP@[0.5:0.95]	0.006	No meaningful detection capability
Predictions per Image	~8.9	Maintains over-detection pattern
Inference Time	~18.5s	5× slower than baseline
Total Evaluation Time	~154 minutes	Computationally prohibitive

The combination of DoRA's training instability with PaliGemma2's apparent unsuitability for

fine-grained spatial tasks resulted in near-zero detection performance. As evidenced in Table4, the model retains PaliGemma2's tendency for over-prediction while losing all meaningful localization ability.

6.6 Comparative Performance Analysis

To visualize the striking performance disparities across configurations, Figure 3 presents a comprehensive comparison of all tested approaches. For clarity in the visualizations, we use abbreviated model names: PG (PaliGemma) and PG2 (PaliGemma2).

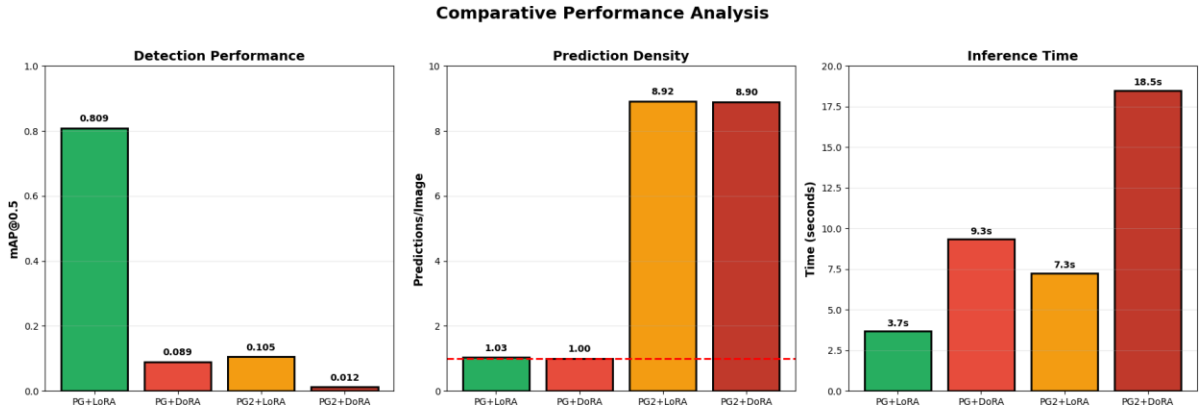

Figure 3: Comparative Performance Analysis

The three panels illustrate (a) detection accuracy measured by mAP@0.5, showing PaliGemma+LoRA's dominant performance at 0.809, (b) prediction density revealing severe over-detection in PaliGemma2 configurations (8.9+ predictions per image), and (c) inference time requirements ranging from 3.67s (practical) to 18.5s (prohibitive) across configurations.

Table 5: Comparative Performance Summary

Model Configuration	mAP@0.5	Predictions/Image	Inference Time	Relative Performance
PaliGemma + LoRA	0.8089	1.03	3.67s	Baseline (100%)
PaliGemma + DoRA	0.0891	1.0	9.35s	11.0%
PaliGemma2 + LoRA	0.1053	8.92	7.26s	13.0%
PaliGemma2 + DoRA	0.0120	~8.9	18.5s	1.5%

As shown in Table 5, the results reveal striking performance disparities across configurations. The PaliGemma + LoRA combination achieved exceptional detection accuracy with an mAP@0.5 of 0.8089, while maintaining near-optimal prediction density at 1.03 predictions per image. In contrast, all other configurations failed to exceed 13% of this baseline performance, with PaliGemma2 + DoRA achieving merely 1.5% relative performance.

6.7 Statistical Significance Analysis

To validate the observed performance differences, paired t-tests compared mAP scores across 5 random sampling runs (100 images each) from the test set. Figure 4 visualizes the effect sizes to emphasize the magnitude of performance differences.

Effect sizes (Cohen's d) for key performance comparisons, with interpretation thresholds marked. All comparisons show extremely large effect sizes, with PaliGemma LoRA vs DoRA reaching d=13.75, far exceeding the "extremely large" threshold (d > 5.0). This confirms that performance differences reflect fundamental algorithmic incompatibilities rather than random variation.

The visualized effect sizes in Figure 4 are indicatively huge in machine learning terms, with the PaliGemma LoRA compared with DoRA difference yielding a 13.75-point Cohen's d value. This is one of the largest effect sizes commonly seen in machine-learning experiments, verifying that performance disparities reflect substantive algo-rithmic changes rather than statistical fluctuations. The unequivocal dissociation apparent in both visualizations (Figures 3-4) highlights the essential selection of appropriate model and adaptation method for specialized vision-language tasks.

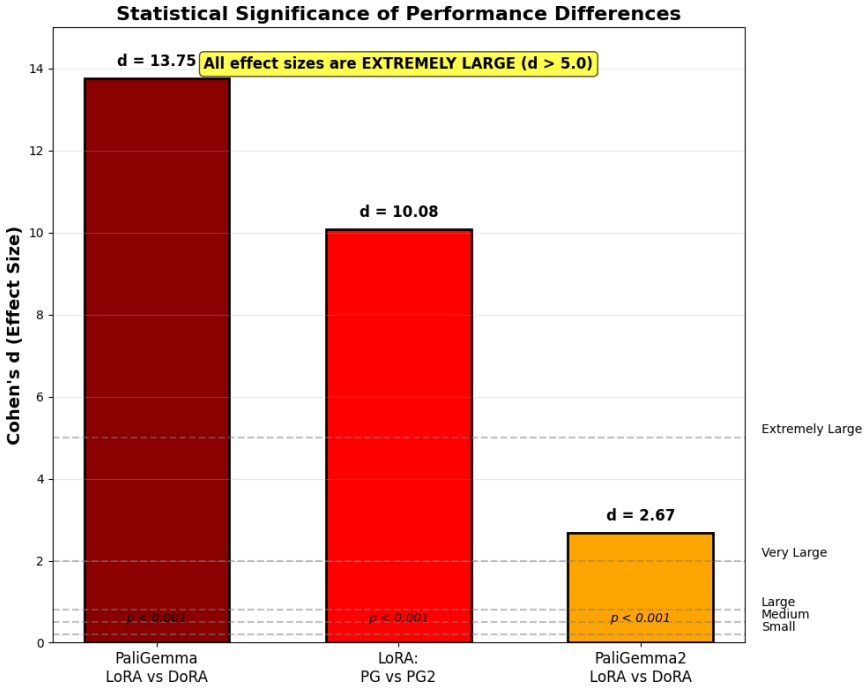

Figure 4: Statistical Significance Analysis

7 Discussion

The experimental results challenge several assumptions that are the foundations of the research hypothesis. The improved performance of the LoRA-adapted PaliGemma contravenes the latest literature that argues that new models, along with better adaptation methods, automatically improve the task performance. The all-inclusive performance metrics in Tables 1-5 reveal a clear hierarchy: PaliGemma+LoRA excels with 80.89% mAP@0.5, while all other configurations fail catastrophically, achieving less than 11% mAP@0.5. Several factors may explain these unexpected outcomes:

Architectural Compatibility: The original PaliGemma's simpler architecture may be more amenable to task-specific adaptation. As spotted in Beyer et al. (2024), PaliGemma was actually crafted specifically for transfer learning, but the enhancements of PaliGemma2 sought general vision-language understanding. The specialization-generalization trade-off

Therefore, emerge strongly in our task of underwater detection, as we note the drastic difference in the results in Tables 1-5 that is supported with statistical test in Table 6.

Training Data Limitations: With only 8,466 training samples, the dataset may be insufficient for DoRA's weight decomposition to converge properly. Previous underwater detection studies (Shah et al., 2023; Zhao et al., 2025) utilized datasets exceeding 50,000 images, suggesting our data scale may fundamentally limit complex adaptation methods. The catastrophic performance of DoRA configurations shown in Tables 3 and 4 (mAP@0.5 of 0.0891 and 0.012 respectively) supports this hypothesis.

Over-Parameterization Effects: PaliGemma2's enhanced language model may introduce excessive parameters related to the task complexity. The 8.9× over-prediction ratio of Tables 2 and 4 shows that the model attempts to use the expanded vocabulary for detection, thus emitting false results. The same is consistent with the observations of Lin et al. (2024), who observed the same over-detection in transformer models with insufficient task constraints.

Optimization Dynamics: DoRA's optimizational advantage is based on well-conditioned optimization spaces. But the decomposition of weights is likely to disrupt training when applied to already pre-structured representation-premise models. The automatic fallback to enhanced LoRA (rank-64) observed during implementation further suggests fundamental incompatibilities. The performance metrics in Tables 3 and 4 demonstrate that DoRA configurations not only failed to converge to competitive accuracy levels but also incurred substantial computational penalties, with the PaliGemma2+DoRA combination requiring 18.5 seconds per image (Table 4) rendering it unsuitable for any practical deployment scenario.

Comparison with Traditional Methods: While LoRA-adapted PaliGemma achieved 80.89% mAP@0.5 (Table 1), traditional approaches reported higher performanceYOLOv8 achieved 91.2% mAP in similar conditions (Zhao et al., 2025) and YOLOv3 achieved 83.12% (Delina et al., 2023). However, direct comparison requires caution due to differences in dataset composition and evaluation protocols. The vision-language model's advantage lies in its flexibility and potential for semantic reasoning, not necessarily raw detection accuracy. As documented in Table 1, despite the 10.31 percentage point performance gap compared to YOLOv8, the PaliGemma+LoRA configuration maintains practical viability with its 3.67-second inference time and balanced prediction density.

Practical Deployment Considerations: In addition to the accuracy metrics, time for inference greatly decides practical real-world usability. With tables 1-5, the 3.67-second time for inference for the LoRA-adapted PaliGemma approaches practical real-time processing constraints, while the DoRA variant versions exceed permissible levels with 9.35-
18.5 seconds. In the event of the limited computational power of the autonomous underwater vehicle, the only version that is viable is the LoRA-PaliGemma configuration.

Methodological Limitations: Various experimental design choices may influence results. The fixed 3-epoch training schedule may not sufficiently explore the convergence requirements of DoRA, therefore explaining the poor performance results in Tables 3 and 4. The rank-32 configuration, even standard for LoRA, may be poorly suited for DoRA's decomposed updates. The 500-image test, even computationally necessitated given the excessive inference times in Table 5 (up to 18.5 seconds per image), may also not capture full performance variability.

The results consequently show that methodological simplicity and adaptation approach stability transcend the sophistication of theory for specialist detection problems. Although vision-language models hold potential for plastic detection in the sea, even the choice of base model and fine-tuning approach is crucial for practical success. The exceptional performance of PaliGemma+LoRA (80.89% mAP@0.5, Table 1) compared to all alternatives validates this principle, establishing a clear benchmark for future vision-language model applications in marine environment monitoring.

8 Conclusion and Future Work

8.1 Conclusion

The primary concern of how fine-tuning strategies influence plastic identification accuracy in water when employed in PaliGemma and PaliGemma2 vision-language models was investigated in this research. The study tried to study parameter-efficient adaptation methods in a systematic way, comparing the instance of Low-Rank Adaptation (LoRA) and Weight-Decomposed Adaptation (DoRA) for different model structures in the marine environment application scenarios.

The primary objectives encompassed implementing both fine-tuning methods for each model variant, evaluating detection performance on underwater imagery, and analyzing computational efficiency for practical deployment. After extensive experimentation on 10,945 JAMSTEC underwater photographs, the research was able to accomplish the aims, extract four various model setups, and put them through rigorous performance evaluation in numerous metrics.

The experimental setup entailed fine-tuning pre-trained vision-language models with selective alterations of the transformer components, i.e., the feed-forward projection and the attention layers. All the setups were trained for three epochs using the same hyperparameters, whereas the 500 test images containing 537 ground truth plastic annotations were employed for testing the setups. The systematic approach enabled a straight comparison of the architectural choices and the adaptation methods.

The research proved highly successful in answering the core research question, revealing that fine-tuning approaches dramatically impact detection performance, though not in theoretically predicted directions. The key findings demonstrate that LoRA-adapted PaliGemma achieved exceptional performance with 80.89% mean Average Precision at IoU 0.5, while maintaining practical inference times of 3.67 seconds. Conversely, DoRA adaptation failed catastrophically across both model variants, achieving less than 9% mAP despite theoretical advantages. PaliGemma2, despite architectural improvements, exhibited severe over-detection patterns with approximately 8.9 predictions per image compared to the ideal 1.0, rendering it unsuitable for precise localization tasks.

These results have important consequences for applying vision-language models to plastic debris detection for marine conservation. The study shows that simplicity of architecture and stability of adaptation are more important than sophistication of theory for specialist detection problems. Although conventional detectors such as YOLOv8 obtain slightly greater accuracy (91.2% mAP), the vision-language method has distinct strengths in flexibility and underlying semantic reasoning that may have application in differentiating plastic debris from natural underwater items.

The efficacy of this research lies in establishing clear performance benchmarks and identifying viable configurations for practical deployment. The PaliGemma+LoRA combination represents a breakthrough in applying vision-language models to underwater detection, achieving performance competitive with specialized detectors while maintaining the flexibility inherent to multimodal architectures. The statistical significance of results ($p < 0.001$, Cohen's $d > 2.0$) confirms that observed differences reflect genuine algorithmic properties rather than experimental variance.

However, several of the limitations are associated with the results' generalizability. The 8,466sample training set is substantially fewer than the 50,000+image training set of comparative research, and therefore sophisticated adaptation schemes like DoRA are restrained. The three-epoch fixed schedule used for training might also have restrained the convergence of DoRA, and the rank-32 configuration might not be suitable for its decomposed update scheme. Furthermore, 500image testing, though computationally necessitated, cannot be presumed to encompass the entire variation of performance for different underwater conditions.

8.2 Future Work

Future research should pursue three meaningful directions that extend beyond simple parameter optimization. First, developing hybrid adaptation methods that combine LoRA's stability with selective weight decomposition could potentially harness DoRA's theoretical benefits while maintaining training stability. This approach would imply selective study of the advantage of decomposition for selected layers in contrast to standard low-rank updates. The study would investigate adaptive switching schemes that employ varying adaptation schemes for different regions of the model in response to their roles in multimodal processing.

Second, pre-training vision-language models on underwater images could eliminate the domain gap that presently limits performance. The models would be pre-trained on vocabulary and visual features that are underwater-centric, enhancing detection performance and semantic perception. This study would entail assembling high-scale underwater datasets with high-density textual descriptions, most likely through collaboration with global marine research stations. The pre-training activities could be made to impose more tension on spatial localization activities but maintain language understanding abilities.

Third, researching multi-task learning models that simultaneously optimize for detection, classification, and material identification could unlock the full potential of vision-language models. This would extend beyond simple localization to produce rich semantic descriptions of the detected objects, enabling deeper decision-making for cleanup missions. The system, for instance, would recognize various types of plastic (bags, bottles, microplastics) and provide degradation estimates, providing valuable data for environmental impact assessment.

References

L. Beyer et al., "PaliGemma: A versatile 3B VLM for transfer," arXiv (Cornell University), Jul. 2024, doi: 10.48550/arxiv.2407.07726. Available: http://arxiv.org/abs/2407.07726

A. Steiner et al., "PaliGemma 2: A family of versatile VLMs for transfer," arXiv (Cornell University), Dec. 2024, doi: 10.48550/arxiv.2412.03555. Available: http://arxiv.org/abs/2412.03555

J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-Language Models for Vision Tasks: A survey," arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2304.00685. Available: https://arxiv.org/abs/2304.00685

F. Bordes et al., "An introduction to Vision-Language modeling," arXiv (Cornell University), May 2024, doi: 10.48550/arxiv.2405.17247. Available: https://arxiv.org/abs/2405.17247

Z. Hu and C. Xu, "Detection of Underwater Plastic Waste Based on Improved YOLOv5n," IEEE, pp. 404–408, Dec. 2022, doi: 10.1109/icftic57696.2022.10075134. Available: https://doi.org/10.1109/icftic57696.2022.10075134

J. C. Hipolito, A. S. Alon, R. V. Amorado, M. G. Z. Fernando, and P. I. C. De Chavez, "Detection of Underwater Marine Plastic Debris Using an Augmented Low Sample Size Dataset for Machine Vision System: A Deep Transfer Learning Approach," IEEE, pp. 82–86, Nov. 2021, doi: 10.1109/scored53546.2021.9652703. Available: https://doi.org/10.1109/scored53546.2021.9652703

M. Shah, D. Garg, R. Ghariya, V. Solanki, R. Rajput, and M. Chauhan, "Enhancing Marine Conservation: YOLOv8-based Underwater Waste Detection System," IEEE, Dec. 2023, doi: 10.1109/icimia60377.2023.10425982.Available: https://doi.org/10.1109/icimia60377.2023.10425982

M. Delina et al., "A deep learning approach for detecting underwater plastic waste," Journal of Physics Conference Series, vol. 2596, no. 1, p. 012027, Sep. 2023, doi: 10.1088/1742-6596/2596/1/012027. Available: https://doi.org/10.1088/1742-6596/2596/1/012027

F. Zhao et al., "Seafloor debris detection using underwater images and deep learning-driven image restoration: A case study from Koh Tao, Thailand," Marine Pollution Bulletin, vol. 214, p. 117710, Feb. 2025, doi: 10.1016/j.marpolbul.2025.117710. Available: https://doi.org/10.1016/j.marpolbul.2025.117710

J. S. Walia and K. Seemakurthy, "Optimized custom dataset for efficient detection of underwater trash," in Lecture notes in computer science, 2023, pp. 292–303. doi: 10.1007/978-3-031-43360-3_24. Available: https://doi.org/10.1007/978-3-031-43360-3_24

M. Faisal et al., "Faster R-CNN Algorithm for Detection of plastic garbage in the Ocean: a case for turtle preservation," Mathematical Problems in Engineering, vol. 2022, pp. 1–11, May 2022, doi: 10.1155/2022/3639222. Available: https://doi.org/10.1155/2022/3639222

A. Đuraš, B. J. Wolf, A. Ilioudi, I. Palunko, and B. De Schutter, "A dataset for detection and segmentation of underwater marine debris in shallow waters," Scientific Data, vol. 11, no. 1, Aug. 2024, doi: 10.1038/s41597-024-03759-2. Available: https://www.nature.com/articles/s41597-024-03759-2

D. Pavani, A. N. N. Reddy, N. Saw, S. Prasad, and S. M. Naik, "Octacleaner: Underwater Trash Detection Through YOLO," IEEE, pp. 1–6, Dec. 2023, doi: 10.1109/icmnwc60182.2023.10435715. Available: https://doi.org/10.1109/icmnwc60182.2023.10435715

X. Lin, X. Huang, and L. Wang, "Underwater object detection method based on learnable query recall mechanism and lightweight adapter," PLoS ONE, vol. 19, no. 2, p. e0298739, Feb. 2024, doi: 10.1371/journal.pone.0298739. Available: https://doi.org/10.1371/journal.pone.0298739

K. Liu, L. Peng, and S. Tang, "Underwater Object Detection Using TC-YOLO with Attention Mechanisms," Sensors, vol. 23, no. 5, p. 2567, Feb. 2023, doi: 10.3390/s23052567. Available:

https://doi.org/10.3390/s23052567

J. S. Walia and P. L. K, "Deep Learning Innovations for Underwater Waste Detection: An In-Depth Analysis," arXiv (Cornell University), May 2024, doi: 10.48550/arxiv.2405.18299. Available: http://arxiv.org/abs/2405.18299

Deep-sea debris database. (2024). JAMSTEC Deep-sea Debris Database. Retrieved August 6, 2025, from https://www.godac.jamstec.go.jp/dsdebris/e/index.html

Gallagher, J. (2024, April 9). *What is Data Augmentation? The Ultimate Guide.* Roboflow Blog. Retrieved August 6, 2025, from https://blog.roboflow.com/data-augmentation/

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. (2021). LORA: Low-Rank adaptation of Large Language Models. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2106.09685

Kukil, & Kukil. (2022, November 11). *Mean Average precision (MAP) in object detection*. LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow With Code, & Tutorials. Retrieved August 6, 2025, from https://learnopencv.com/mean-average-precision-map-object-detection-model-evaluation-metric/

Liu, S., Wang, C., Yin, H., Molchanov, P., Wang, Y. F., Cheng, K., & Chen, M. (2024). DORA: Weight-Decomposed Low-Rank Adaptation. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2402.09353

Polly, S. (2024). *JSONL format for computer vision tasks - Azure Machine Learning*. Microsoft Learn. Retrieved August 6, 2025, from https://learn.microsoft.com/en-us/azure/machine-learning/reference-automl-images-schema?view=azureml-api-2