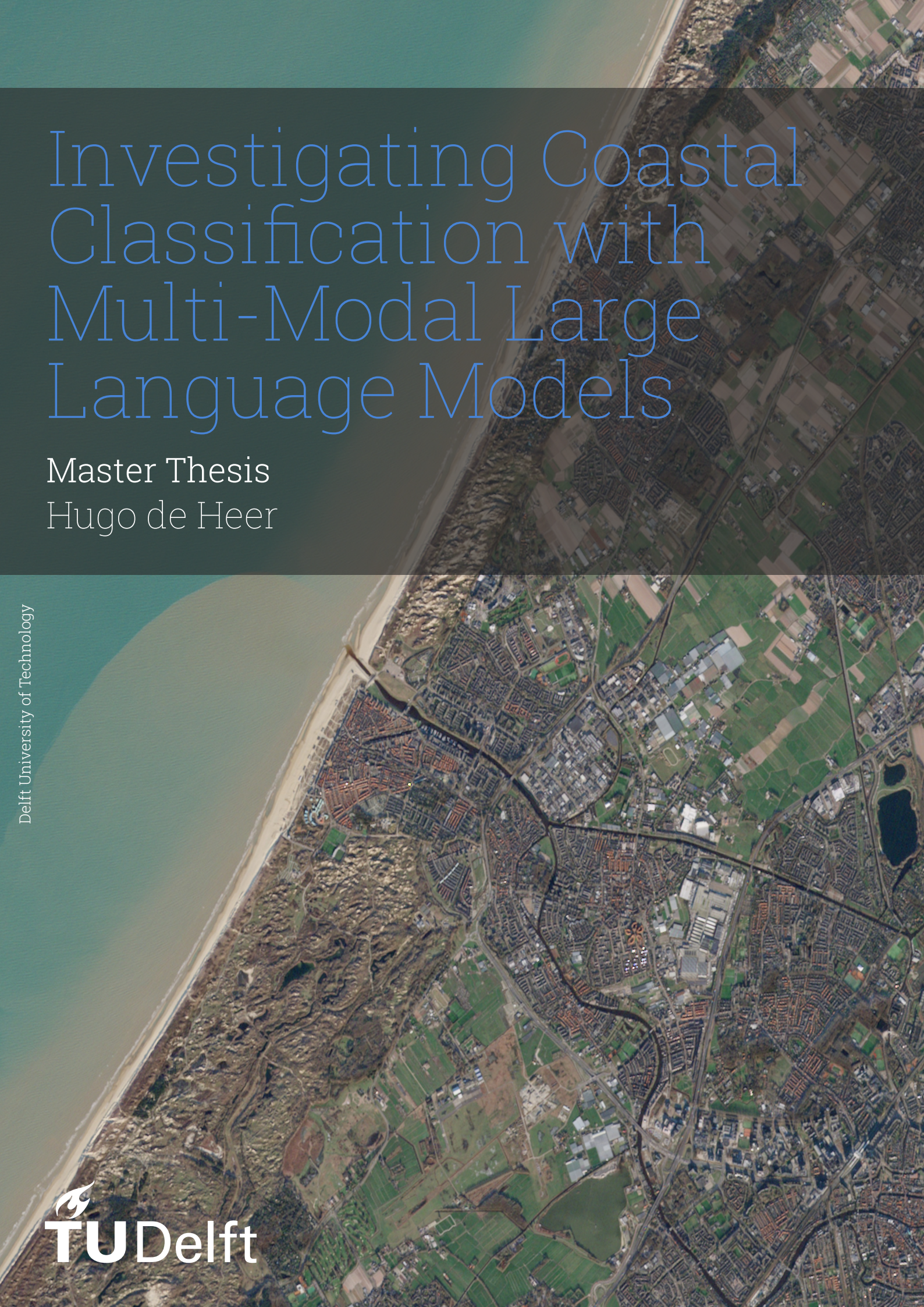
Investigating Coastal Classification with Multi-Modal Large Language Models

Master Thesis

Hugo de Heer



Delft University of Technology

TU Delft

Investigating Coastal Classification with Multi-Modal Large Language Models

by

Hugo de Heer

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defined publicly on the 4th of June, 2025.

Student number:	4953398	
Project duration:	September 2024 - June 2025	
Thesis committee:	Dr. ir. J.C. van Gemert,	TU Delft, Thesis Advisor
	MSc F.R. Calkoen,	Deltares, Daily Supervisor
	Dr. ir. A.M. Moreno-Rodenas,	Deltares, Daily Supervisor
	Dr. ir. S. Verwer,	TU Delft, External Committee Member
Faculty:	Faculty of Computer Science, Delft	

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

This thesis, titled *"Investigating Coastal Classification with Multi-Modal Large Language Models"*, presents the results of my research for the Master of Science degree in Computer Science at TU Delft. Conducted in collaboration with Deltares, this work began in September 2024 and concluded in May 2025, marking a period of interesting research, experimentation, and writing.

It is fascinating to note how rapidly AI is developing in this era of research. Every month, new state-of-the-art AI models are being released, requiring a dynamic approach of performing research. Furthermore, this is also inspiring to observe how far we can push AI to solve important problems in the real world. I am positive that this thesis marks the start of the use of large language models in the coastal field. This research was performed within the Computer Vision Laboratory at the Technical University of Delft in collaboration with Deltares. Under the supervision of Dr. ir. J.C. van Gemert from TU Delft and the daily supervision of Deltares researchers Dr. ir. A.M. Moreno-Rodenas and MSc F.R. Calkoen.

I would like to express my deep gratitude to my Deltares supervisors. To MSc F.R. Calkoen for the insightful coastal-themed brainstorming sessions and assisting in developing the dataset. Dr. ir. A.M. Moreno-Rodenas, for his structured approach of doing research and invitations to external activities and projects within Deltares which were also a great and enjoyable learning experience. Thank you both for that. I would also like to thank Dr. ir. J.C. van Gemert for the helpful guidance and insights in the meetings that we had. And lastly, thanks to Dr. ir. S. Verwer for his interest in joining the thesis committee.

I hope this work inspires further exploration at the intersection of AI and environmental science.

Hugo de Heer
Delft, May 2025

Nomenclature

Abbreviations

Abbreviation	Definition
CNN	Convolutional Neural Network
COP DEM GLO	Copernicus Global Digital Elevation Model
DeltaDTM	Deltares Digital Terrain Model
DEM	Digital Elevation Model
LLM	Large Language Model
MLLM	Multi-modal Large Language Model
MLP	Multi-Layer Perceptron
MNDWI	Modified Normalized Difference Water Index
NDMI	Normalized Difference Moisture Index
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
NIR	Near-Infrared
OSM	OpenStreetMap
RGB	Red Green Blue
ResNet	Residual Neural Network
SOTA	State Of The Art
SWIR	Short-wave Infrared
V1	Prompting variant using a structured zero-shot technique
V1.1	Prompting variant V1 extended with reasoning capabilities
V2	Prompting variant using a structured few-shot technique
VLM	Vision Language Model
VQA	Visual Question Answering
CoT	Chain-of-Thought

Symbols present in the scientific paper

Symbol	Definition	Unit
c	Coastal type class	[-]
s	Shore type class	[-]
b	Visiblity of built environment	[-]
d	Visibility of coastal defences	[-]
C	Coastal Context for MLLM models	[-]
Q	Classification questions asked to MLLM models	[-]
A	Predictions of MLLM models	[-]
I	Images given to MLLM models	[-]
R	Reasoning part of prompt	[-]
F_1	Average F1-score over all labels	[-]
$F_1(c)$	F1-score of the coastal classification (8 classes)	[-]
$F_1(s)$	F1-score of the shore classification (5 classes)	[-]
$F_1(b)$	F1-score of built environment classification (binary)	[-]
$F_1(d)$	F1-score of coastal defences classification (binary)	[-]

Contents

Preface i

Nomenclature ii

1 Introduction 1

2 Background 3
2.1 The Coastal Environment . . . 3
2.1.1 Coastal and Shore Types . . . 4
2.1.2 Built Environments and Coastal Defences . . . 6
2.2 Dataset Features . . . 7
2.2.1 Data Sources . . . 7
2.2.2 Feature Modalities . . . 7
2.2.3 Feature Summary . . . 9
2.3 Deep Learning . . . 10
2.3.1 Perceptrons and Neural Networks . . . 10
2.3.2 Multi-Layer Perceptrons and Activation Functions . . . 10
2.3.3 Convolutional Neural Networks . . . 10
2.3.4 Residual Neural Networks . . . 11
2.3.5 Large Language Models . . . 12
2.3.6 Multimodal Large Language Models . . . 12
2.4 Model Training and Optimisation . . . 14
2.4.1 Training Methodology . . . 14
2.5 Prompting Techniques for LLMs . . . 16
2.5.1 Zero-shot prompting . . . 16
2.5.2 Few-shot prompting . . . 16
2.5.3 Chain-of-Thought Prompting . . . 17
2.5.4 Structured Output Formatting . . . 17

3 Scientific Paper 18

References 19

1

Introduction

The coastal zone is a dynamic interface where land meets the ocean, encompassing diverse ecosystems such as beaches, dunes, estuaries, and deltas. This zone is highly productive, both ecologically and economically, providing critical ecosystem services, supporting high biomass production and biodiversity, and serving as the first line of defence against coastal hazards such as flooding [1]. The coastal zone is continuously reshaped by natural forces, including wave action, tidal fluctuations, and wind patterns, which contribute to its ever-changing landscape [2].

However, coastal zones are increasingly under pressure from accelerating climate change [3] and anthropogenic factors [4]. The frequency and intensity of coastal flooding events and the rate of coastal erosion are on the rise due to the changing climate, therefore posing a significant threat to coastal communities [5]. Projections indicate that without adaptation, a 0.75-meter rise in sea level would double the present at-risk population [6]. Furthermore, the rate of sea-level rise itself has been accelerating each year over the past century [7]. This requires a timely and adequate commitment to adaptation. Effective adaptation relies on informed decision-making, which in turn depends on robust and accessible data. In recent years, historical satellite image catalogues and cloud computing platforms have provided an unprecedented view of the coast [8]. However, making sense of this growing volume of data remains a challenge. Artificial Intelligence (AI) offers powerful tools to automate and enhance the extraction of relevant information from remotely sensed data. Despite this potential, the use of AI in coastal science remains largely untapped [9].

AI has rapidly evolved in recent years, particularly in the areas of computer vision, and natural language processing [10]. A prime example in environmental monitoring is Google's Dynamic World, which utilises satellite imagery and AI in the form of Convolutional Neural Networks (CNNs) to provide real-time environmental monitoring [11]. In the field of language, large language models (LLMs) such as GPT-3 [12] have demonstrated power in understanding and generating the human language, opening up new possibilities for tasks such as question answering and translation. Building on these advancements, multi-modal large language models (MLLMs) were introduced in 2021 [13] to combine visual and textual data, offering deeper insights by integrating high-level reasoning with both imagery and language.

Classifying coasts and shores into clear distinct types can assist in monitoring environmental changes, disaster risk management and the preservation of ecosystems [14, 15, 16]. Traditional CNN-based models are limited to processing strictly imagery, extracting patterns from unstructured pixels without leveraging external knowledge. In contrast, MLLMs go beyond pure visual encoding. Namely, they integrate pre-trained knowledge, language-style reasoning, and high-level imagery interpretation. This enables these models to not only classify coastal environments but also provide richer, more contextual insights in natural language. This enhances accessibility for policymakers, environmental managers, and local communities, who may lack technical expertise but require actionable insights for decision-making in coastal monitoring and management. However, it remains crucial that these users critically evaluate AI model outputs, considering potential limitations and uncertainties.

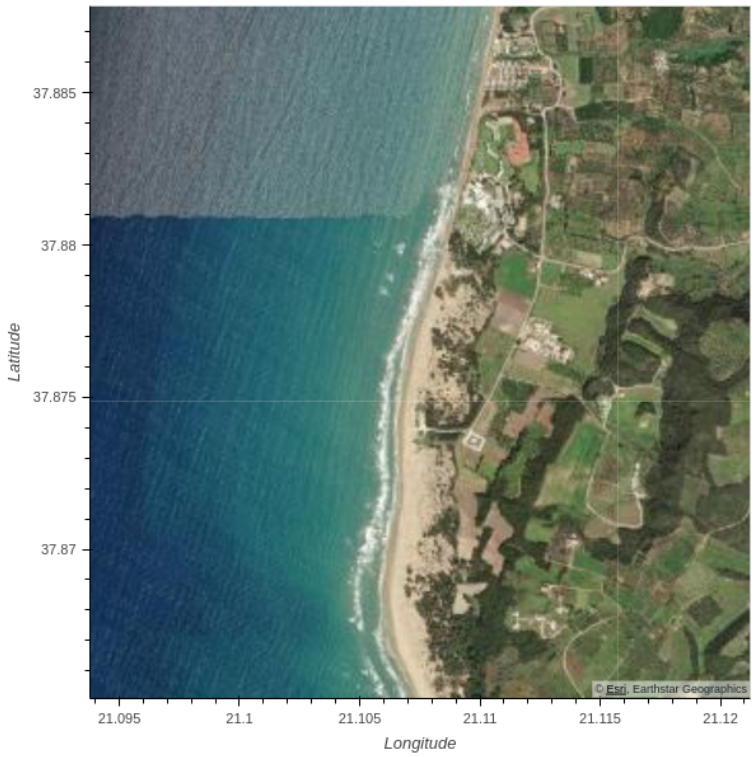


Figure 1.1: Example RGB image of a coastal area that is present in the coastal classification dataset. The image depicts a coastline located in Greece, near the Ionian Sea.

This thesis is organized as follows:

- Background material: An additional section aimed at making the concepts in the research article more accessible to the interested reader.
- The research article: The core of this report with the main results in a format acceptable for a suitable computer vision conference.

Background. The background chapter contains all the supplementary materials needed to understand the concepts mentioned in the research article, including an overview of the coastal environment and deep learning in the field of computer vision. Furthermore, it explains the type and architectures of the multi-modal models used in the research article and details the custom dataset used for this research.

Research article. This study investigates the extent to which automated coastal classification using remote sensing data can be improved by adopting multi-modal large language models (MLLMs). It explores the role of MLLMs in enhancing coastal classification by integrating satellite imagery with additional contextual information. More specifically, we use a novel coastal classification dataset [17] to investigate the following research questions:

1. How does a traditional CNN approach compare to untrained MLLMs in coastal classification?
2. What is the impact of different prompting techniques on the classification performance of MLLMs?
3. How can additional satellite bands, beyond RGB, be effectively embedded into an MLLM for coastal classification?
4. Which features contribute most to the performance differences observed across models and approaches?

2

Background

This chapter introduces the foundational concepts, terminology, and technical components relevant to this research. It is intended to provide context and clarity for readers across diverse disciplines, enabling a better understanding of the methodology and results presented later in the thesis.

We begin with an overview of the coastal environment in Section 2.1, including definitions and distinctions between different coastal and shore types. This environmental context is followed in Section 2.2 by a specification of the dataset feature maps used in the classification task, such as spectral bands, elevation data, and derived indices.

Section 2.3 introduces key machine learning principles, including both computer vision techniques and multimodal learning with large language models. It highlights the architectures and models used in the study, with a focus on convolutional neural networks and transformer-based models.

In Section 2.4, we examine the training and optimization strategies employed to enhance model performance, as well as the prompting techniques applied to large multimodal models for interpreting coastal scenes. Altogether, this background forms the basis for understanding the interdisciplinary approach of this thesis.

2.1. The Coastal Environment

The coastal zone marks the dynamic interface between land and sea, shaped by geomorphological processes and human interventions. It is a spatially complex and heterogeneous environment [18]. While coastal classifications have traditionally relied on expert interpretation and field-based geomorphology [19, 20], recent advances in satellite imagery and AI enable a more scalable, image-based perspective [21, 22]. In this study we follow Calkoen et al. (2021) [17] to characterise the coast by four different dimensions to capture the complexity of the coast in a form that supports machine understanding:

1. **Coastal type**: The broader geomorphological form of the coast (e.g., dunes, cliffs), including a class *engineered structures* for areas where the geomorphology is no longer active.
2. **Shore type**: The predominant surface composition near the waterline (e.g., muddy sediments, rocky platforms),
3. **Built environment presence**: A binary label indicating visible human infrastructure,
4. **Coastal defence presence**: A binary label marking protective structures such as sea walls or groynes.

These four dimensions help structure the complexity of the coast into separate parts that can be observed from space. They combine information about natural landforms, surface materials, and human influence. Together, they form the basis of the classifications we aim to automatically recognise using satellite images. The next sections explain each of these labels in more detail and describe how they relate to the coastal images used in this research.

2.1.1. Coastal and Shore Types

Figure 2.1: Examples of all coastal types used in this study. From top-left to bottom-right: Cliffed or Steep, Moderately Sloped, Dune, Sediment Plain, Wetland, Inlet.

To describe the physical appearance of the coastline in a structured way, we distinguish between *coastal types* and *shore types*. These two characteristics refer to different spatial scales and physical properties of the coast. Coastal systems respond very differently based on their cross-shore variability; i.e., a sandy shore backed by a cliffed coast will respond much differently than a sandy shore on a dune coast. The classification that is followed in this study aims to capture that cross-shore variability [17].

Coastal types describe the broader geomorphology of the coast. Namely, its shape, slope, and dominant surface material. For example, some coastal areas are steep and rocky, forming cliffs, while others are flat and sandy, forming wide beaches or sediment plains. In this study, we use the following coastal types [17]:

- **Cliffed or Steep**: Coastal areas with cliffs or steep rock faces, with slopes of 30 degrees or greater.
- **Dune**: Sandy coastal areas characterised by wind-formed dunes, often stabilised by vegetation

such as grasses.

- **Bedrock Plain**: Low-lying coastal areas (<15m) primarily formed by consolidated bedrock, including skerries, with minimal elevation change.
- **Engineered Structures**: Coastal areas dominated by engineered structures such as port areas, sea walls and dams, where the natural coastal landscape is obscured or heavily modified.
- **Inlet**: Narrow coastal waterways where the sea meets the land, creating dynamic systems such as estuaries, tidal bays, and lagoons.
- **Moderately Sloped**: Coastal areas with gentle to moderate slopes (<30°), typically composed of unconsolidated sediment or soft rock.
- **Sediment Plain**: Low-lying coastal areas (<15m) with flat or gently sloping unconsolidated sediment, often featuring beach ridges or washover complexes.
- **Wetland**: Coastal areas periodically flooded, including environments such as tidal flats, salt marshes, mangroves, sabkhas, and peatlands.

Examples of all the coastal types can be observed in Figure 2.1.

Figure 2.2: Examples of all shore types used in this study. From top-left to bottom-right: Sandy Gravel or Small Boulders, Muddy Sediments, Rocky Shore Platform, Ice or Tundra, No Sediment or Shore Platform.

Shore types describe what the coast consists of right at the waterline. This includes the material that is visible on the beach or intertidal zone, which affects how waves interact with the land. The following categories are used:

- **Sandy Gravel or Small Boulder Sediments**: Shorelines composed of unconsolidated materials such as sand, gravel, shingles, and small boulders (0.0652 to 512 mm in diameter).
- **Muddy Sediments**: Shorelines dominated by fine-grained sediments like silt and clay, forming environments such as mudflats and tidal flats.

- **Rocky Shore Platform or Large Boulders**: Shorelines composed of solid rock formations, including shore platforms or large boulders greater than 512 mm in diameter.
- **Ice or Tundra**: Shorelines characterised by icy environments or tundra, including frozen sea edges, icebergs, and glaciers, typically found in polar regions.
- **No Sediment or Shore Platform**: Shorelines with minimal visible sediment, typically around rocky cliffs, steep faces, or human-made structures such as sea walls.

2.1.2. Built Environments and Coastal Defences

In addition to natural landforms, coastal areas are also classified based on the visible presence of human infrastructure and structures that protect against coastal erosion and flooding. These features are highly relevant for understanding coastal exposure and associated risks.

Built environment refers to the presence of human-made structures such as buildings, industrial zones, roads, or port infrastructure. These are identified as part of the satellite image and classified as either:

- **Present**: The coastal area is characterised predominantly by human-made structures, including buildings, industrial complexes, and port facilities.
- **Absent**: The coastal area remains largely natural, with minimal or no presence of built structures like buildings, industrial zones, or ports.

Coastal defences refer to structures specifically designed to protect land from erosion and flooding, such as sea walls, breakwaters, and revetments. These are also classified as:

- **Present**: Visible hard engineering structures, designed to protect against coastal erosion and flooding (e.g., sea walls, breakwaters), are present.
- **Absent**: No visible hard engineering structure, designed to protect against coastal erosion and flooding (e.g., sea walls, breakwaters), are present.

Figure 2.3 shows example combinations of these two labels.

Figure 2.3: Examples of coastal areas with different combinations of built environment and defence presence: (1) no built, no defence; (2) no built, defence present; (3) built, no defence; (4) built and defence present.

2.2. Dataset Features

The classification models in this thesis operate on a 12-channel input representation derived from satellite and elevation data. These channels capture optical reflectance, surface topography, and spectral indices commonly used in remote sensing. This section introduces the origin, structure, and meaning of each feature.

2.2.1. Data Sources

The dataset combines optical satellite imagery and digital elevation data:

- **Sentinel-2 imagery:** Provided by the European Space Agency (ESA) [23], Sentinel-2 offers multispectral images at 10-60 meter resolution. The dataset uses 6 spectral bands from Sentinel-2 Level-2A surface reflectance data, including visible (Blue, Green, Red), near-infrared (NIR), and shortwave infrared (SWIR).
- **Copernicus DEM (COP DEM GLO):** A global digital elevation model with 30-meter resolution, produced by the Copernicus programme [24].
- **DeltaDTM:** A high-resolution digital terrain model developed by Deltares, a lidar-based improvement of Copernicus DEM. It is particularly relevant for coastal regions, where absolute elevation differences are relatively small. [25].

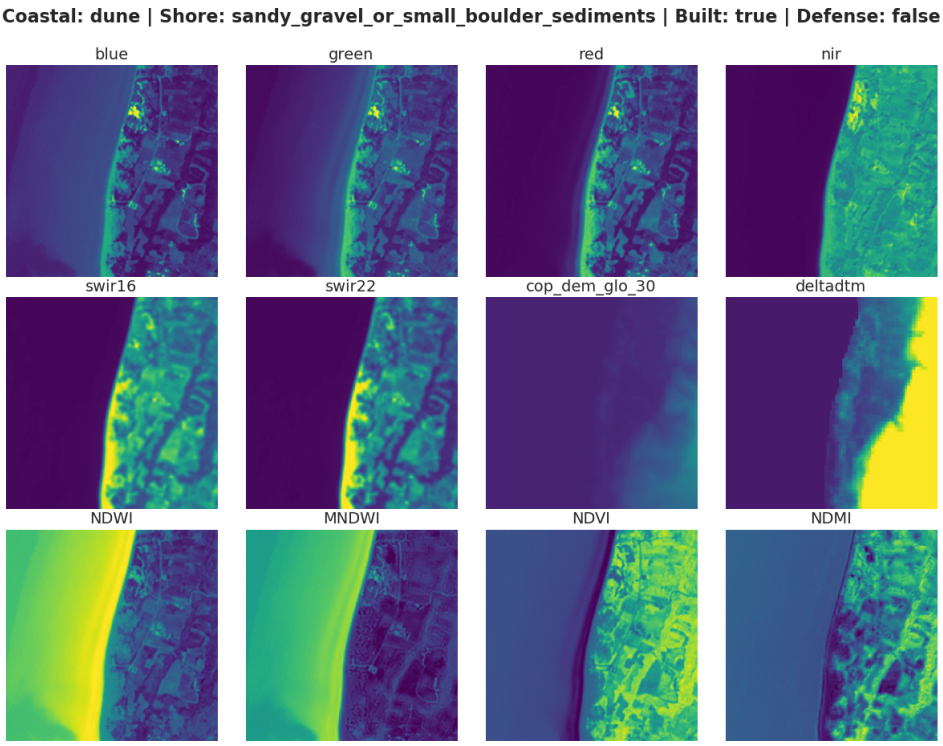


Figure 2.4: Example image containing all 12 channels of a coastal transect with a Dune coastline, Sandy Gravel or Small Boulder Sediments shore type and containing a built environment with no coastal defences. Colour intensity reflects relative values within each channel, namely brighter regions indicate higher values, while darker areas represent lower ones. For example, the dune system is clearly visible in the DeltaDTM (bright yellow indicating higher elevation) but not in the coarser Copernicus DEM. Likewise, built-up areas exhibit low NDMI values (dark regions), indicating minimal moisture content compared to surrounding vegetated or wet areas.

2.2.2. Feature Modalities

The 12 channels fall into three categories:

Spectral Reflectance Bands (6 channels)

These bands represent surface reflectance in different parts of the electromagnetic spectrum:

- **Blue (490 nm)** - Highlights water bodies and detects suspended sediments. .

- **Green (560 nm)** - Sensitive to chlorophyll and therefore to vegetation
- **Red (665 nm)** - Enhances vegetation contrast.
- **NIR (842 nm)** – Sensitive to vegetation structure and biomass.
- **SWIR16 (1610 nm)** – Responds to soil moisture and dry vegetation.
- **SWIR22 (2190 nm)** – Useful for sediment discrimination.

Elevation Models (2 channels)
These channels provide topographic context:

- **COP DEM GLO** – Raw elevation values at 30m resolution.
- **DeltaDTM** – A terrain-refined elevation model specifically tailored to coastal morphology.

Spectral Indices (4 channels)
These are normalized ratios derived from spectral bands, designed to highlight specific surface characteristics:

- **NDVI (Normalized Difference Vegetation Index)**

$$NDVI = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}}$$

  NDVI increases in proportion to vegetation growth [26].
- **NDWI (Normalized Difference Water Index)**

$$NDWI = \frac{\text{Green} - \text{NIR}}{\text{Green} + \text{NIR}}$$

  Highlights open water bodies [27].
- **MNDWI (Modified NDWI)**

$$MNDWI = \frac{\text{Green} - \text{SWIR16}}{\text{Green} + \text{SWIR16}}$$

  More effective at separating water from built-up land and vegetation [28].
- **NDMI (Normalized Difference Moisture Index)**

$$NDMI = \frac{\text{NIR} - \text{SWIR16}}{\text{NIR} + \text{SWIR16}}$$

  Indicates vegetation water content and soil moisture [29].

These derived indices augment the raw reflectance values with features that emphasise vegetation, moisture, and land-water contrast, which are key elements in coastal zone classification.

2.2.3. Feature Summary

An overview of all 12 channels used in the classification model is given below:

Channel	Description
Blue	Sentinel-2 Band 2 (490 nm)
Green	Sentinel-2 Band 3 (560 nm)
Red	Sentinel-2 Band 4 (665 nm)
NIR	Sentinel-2 Band 8 (842 nm)
SWIR1	Sentinel-2 Band 11 (1610 nm)
SWIR2	Sentinel-2 Band 12 (2190 nm)
COP DEM GLO	Copernicus Global DEM (30m)
DeltaDTM	Deltares Digital Terrain Model
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
MNDWI	Modified Normalized Difference Water Index
NDMI	Normalized Difference Moisture Index

Table 2.1: Overview of the 12 feature channels used in the classification dataset.

2.3. Deep Learning

This section introduces the key machine learning concepts used in this study, from the fundamentals of neural networks to modern large multimodal models. These concepts are essential to understand how both computer vision and language-based AI can support coastal classification from satellite imagery.

2.3.1. Perceptrons and Neural Networks

Artificial neural networks are built from basic computational units called perceptrons. A perceptron takes multiple inputs $x_1, x_2, \ldots, x_n$, multiplies each with a corresponding weight w_i, adds a bias term b, and passes the result through an output function. Mathematically:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

where f is often a non-linear activation function (which is discussed in section 2.3.2). Neural networks can be seen as nested mathematical formulas, composed of many layers of such transformations, enabling the system to model patterns that are present in all types of data.

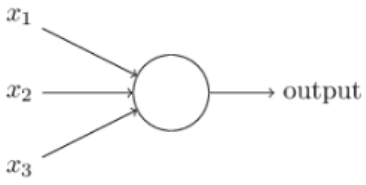


Figure 2.5: Example of a perception that takes 3 inputs x_1, x_2, x_3 and outputs y.

2.3.2. Multi-Layer Perceptrons and Activation Functions

When multiple perceptrons are stacked in layers, it forms a Multi-Layer Perceptron (MLP): a feedforward neural network as seen in Figure 2.6. Each layer transforms its input into a more abstract representation. However, we can observe that the operations done in every layer of the perception are linear. Stacking linear operations alone would collapse to a single linear mapping. To model the complex, non-linear patterns in real-world data, activation functions such as ReLU [30] or Sigmoid [31] are essential. These introduce non-linearities, allowing networks to approximate arbitrary functions, and allowing for pattern modelling of all types of data.

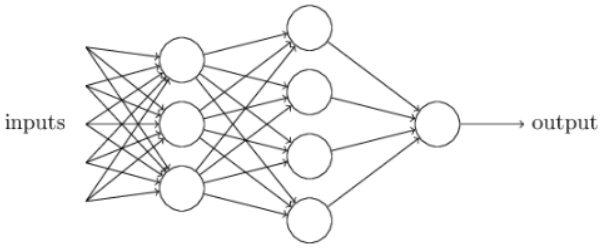


Figure 2.6: Example of a Multi-Layer Perceptron with 3 layers.

2.3.3. Convolutional Neural Networks

For image data, Convolutional Neural Networks (CNNs) are more effective than MLPs because they explicitly exploit the spatial structure of images [32]. Rather than treating an image as a flat vector of pixel values, CNNs preserve the two-dimensional layout and apply learnable filters to detect local patterns.

The core operation of a CNN is the convolution. A convolution involves sliding a small matrix called a kernel or filter across the input image. At each spatial location, the kernel performs an element-wise multiplication with the input patch and sums the result. This sum becomes a single pixel in the output

feature map. Formally, given a 2D input I and a kernel K of size $m \times m$, the convolution output O at position (i, j) is:

$$O(i,j) = \sum_{u=0}^{m-1} \sum_{v=0}^{m-1} I(i+u, j+v) \cdot K(u,v)$$

This operation is repeated over the entire input, shown in Figure 2.7 allowing the network to learn spatially localised patterns, such as edges, textures, or shapes, by learning the kernel weights during training. Importantly, the same kernel is reused across the image, drastically reducing the number of parameters compared to fully connected networks and enabling translation invariance.

Multiple kernels are typically applied in each convolutional layer, resulting in several output channels (also called feature maps). These are followed by non-linear activation functions (such as ReLU [30]) and pooling layers for downsampling (such as Max Pooling [33]).

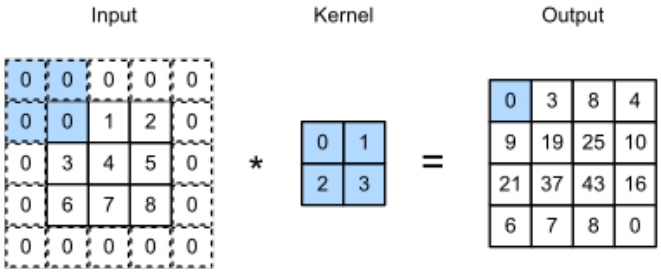


Figure 2.7: Convolution operation on a 2D 5x5 input matrix with a 2x2 kernel.

In this study, CNNs are used as a baseline model trained directly on satellite images, both using RGB and multispectral input data, to classify multiple coastal attributes.

2.3.4. Residual Neural Networks

As neural networks become deeper (having more layers), training becomes more difficult. A key reason for this is because of the vanishing gradient problem: during backpropagation, gradients are propagated from the output layer back to the earlier layers. However, in deep networks, these gradients often shrink exponentially as they move backwards. As a result, early layers receive minuscule updates in their trainable parameters, effectively supressing learning in those layers. This phenomenon leads to poor training performance, even though deeper networks theoretically have greater capacity to learn complex patterns.

Residual Networks (ResNets) address this by introducing skip connections where the input x is added with the learned residual mapping $F(x)$ as shown in Figure 2.8. These skip connections allow gradients to flow more directly through the network, mitigating the vanishing gradient issue. If any layer ends up hurting the performance of the model, it can be skipped due to the presence of the skip-connections.

$$\text{Output} = F(x) + x$$

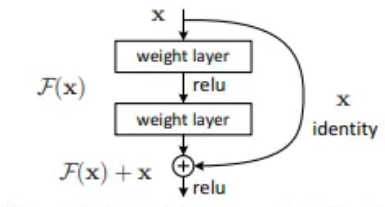


Figure 2.8: Residual learning: a building block [34].

ResNet50

The ResNet50 architecture used in this study is a CNN consists of 50 layers organised into blocks with residual connections. Its architecture can be observed Figure 2.9 below.

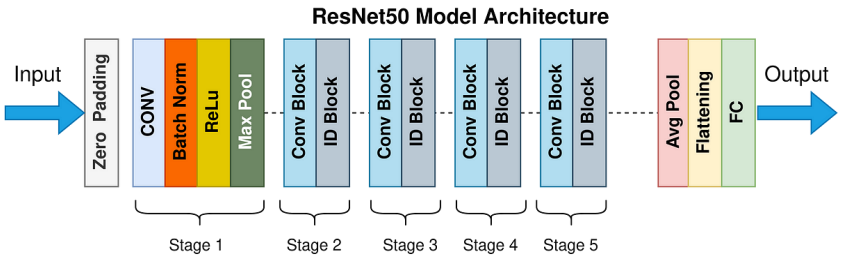


Figure 2.9: Model architecture of ResNet50.

2.3.5. Large Language Models

Transformers are the foundational architecture behind modern Large Language Models (LLMs). Unlike CNNs and Residual Neural Networks (ResNets), which were originally designed for vision and spatial feature extraction, transformers were developed specifically to handle the challenges of modelling relationships in sequences such as text. Traditional sequence models such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) struggle with modelling long-range dependencies and parallelization [35].

Transformers overcome these limitations with a novel mechanism called self-attention [35]. In each layer, every token attends to every other token, weighted by their learned relevance. This is implemented using scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Where:

- Q, K, and V are the queries, keys, and values, obtained via learned linear projections of the input.
- d_k is the dimensionality of the keys.

This design enables the model to dynamically focus on relevant parts of the sequence, regardless of their distance. It also enables capturing long-distance dependencies, which is important when aiming to capture patterns in high-dimensional data.

Large Language Models (LLMs) such as GPT-3 [12] and LLaMA [36] are transformer-based models trained on massive text datasets. These models can generate, summarise, translate, and perform complex language-based tasks that resemble reasoning. But transformers can also be extended to process vision and language jointly, a key idea in this thesis.

2.3.6. Multimodal Large Language Models

Multimodal learning refers to models that combine different types of input, for example text and images. For coastal classification, satellite imagery offers visual cues, while external metadata (e.g., place names, descriptions) can add semantic context. Multimodal models process and align these inputs in a shared latent space, enabling richer understanding than either modality alone.

Multi-modal Large Language Models (MLLMs) are advanced transformer architectures that take images as input alongside natural language prompts and produce text-based outputs. They are pretrained on massive image-text datasets and fine-tuned for tasks such as image captioning, visual question answering, and multimodal classification [37, 38]. In this study, we evaluate three MLLMs with distinct architectural and access characteristics:

Qwen2.5-VL-Instruct-7B

Qwen2.5-VL-Instruct-7B is the latest leading model of the Qwen vision-language series [39], containing approximately 7 billion trainable parameters. It is an open-source multimodal model developed by Alibaba, released in 2025. The model processes visual and textual inputs jointly and is optimised for structured prompting tasks. It supports high-resolution image inputs and returns detailed, constrained outputs.

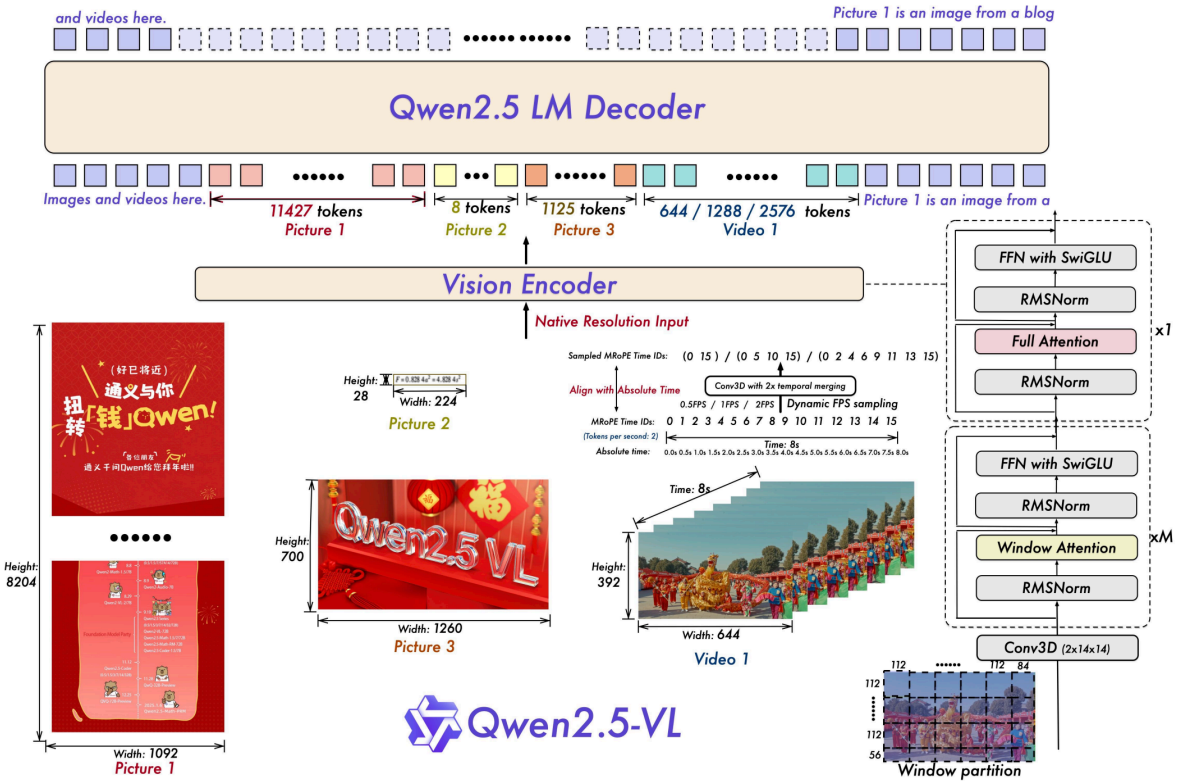


Figure 2.10: Architecture overview of Qwen2.5-VL, a multimodal vision-language model. The system supports input from high-resolution images, text, and videos, which are encoded by a Vision Encoder (Vision Transformer) using native resolutions and 3D convolution with temporal merging. Visual tokens are aligned with absolute time and mapped to token sequences for the Language Model (LM) Decoder. The decoder employs layers with SwiGLU [40] feed-forward networks, full or windowed attention, and RMS normalisation [41]. This enables the model to handle various input modalities while maintaining spatial-temporal structure. This figure is adopted from Alibaba's Qwen2.5-VL technical report. [39]

Llama-3.2-11B-Vision-Instruct

Llama-3.2-11B-Vision-Instruct is a multimodal extension of Meta's LLaMA 3.2 language model family, designed to jointly process visual and textual inputs. It contains approximately 11 billion trainable parameters and builds upon the LLaMA 3.2 transformer backbone, which is an architecture that incorporates improvements in tokeniser efficiency, normalisation schemes, and pretraining stability.

The visual processing module follows a multi-stage design and is based on a Vision Transformer (ViT), which encodes input images into patch embeddings. To enrich the visual representation, the model aggregates not only the final output of the vision encoder but also five intermediate hidden states drawn from different layers within the visual backbone. These multi-level embeddings are projected into the same token space as the language model and then concatenated with text tokens to form a unified input sequence. This sequence is passed to the standard transformer decoder, enabling the model to perform joint reasoning over both modalities.

GPT-4o

GPT-4o is OpenAI's latest multimodal leading model, capable of processing high-resolution images and structured classification prompts. It offers state-of-the-art reasoning and generalisation across modalities [42]. Although OpenAI has not disclosed the exact number of parameters, external analyses

estimate its active parameter count to range between 100 and 300 billion [43]. GPT-4o operates as a black-box system without access to internal gradients or model weights, which limits fine-tuning or interpretability.

2.4. Model Training and Optimisation

This chapter explains the core training techniques and optimisation strategies used in the scientific study. Training methods mentioned in the main paper, such as early stopping, cross-validation, and data augmentation, are clarified here.

2.4.1. Training Methodology

During training, various strategies and methods have been applied, which are explained in the sections below.

Cross-validation

Cross-validation is a robust model evaluation technique that tests how well a model generalises to unseen data. Instead of training on a single split, the dataset is divided into multiple parts (called folds), and the model is trained and validated multiple times, each time with a different fold held out as validation. In this study, a 5-fold group cross-validation was used.

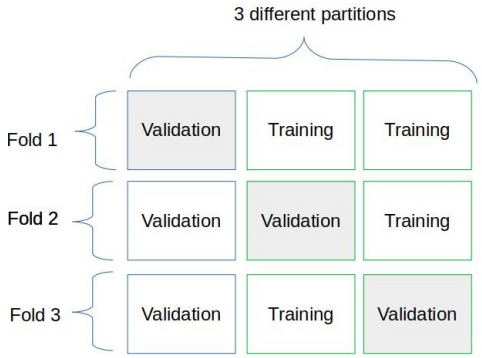


Figure 2.11: Example of a 3-fold Cross-validation setup.

Learning rate scheduling

The learning rate controls how much a model updates its parameters after each training step. A learning rate that is too high can overshoot optimal values, while one that is too low can slow down learning. Scheduling techniques adjust the learning rate dynamically during training, typically starting high and reducing it gradually. This helps the model converge more smoothly. Figure 2.12 shows an example of a cosine annealing scheduler, which is also the one used in this thesis.

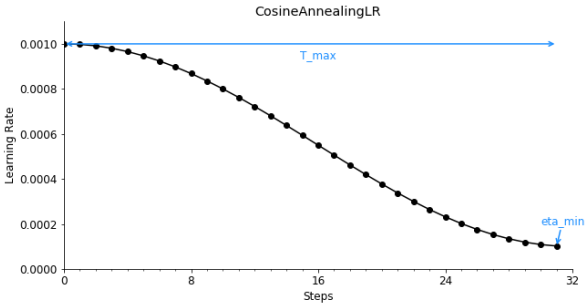


Figure 2.12: Visualisation of a Cosine Annealing Learning Rate (LR) schedule. The learning rate begins at its initial value (0.0010) and gradually decreases to a minimum value (`eta_min`) following a cosine curve over a predefined number of steps (`T_max`). This scheduling strategy enables large updates at the beginning of training and progressively smaller steps as convergence is approached.

Early stopping and overfitting

Early stopping is a regularisation technique used to prevent overfitting during model training. Overfitting occurs when a model learns the training data too well, including its noise or random fluctuations, which harms its ability to generalise to unseen data. This is visible when training performance continues to improve, but validation performance begins to degrade. This is a sign that the model is memorising rather than learning. Regularisation refers to a set of techniques designed to limit model complexity and improve generalisation. These techniques help the model focus on the underlying patterns in the data instead of overfitting to specific examples. In early stopping, the model is evaluated on a validation set after each training epoch. If the validation performance does not improve after a predefined number of epochs (also called the patience), training is halted. This captures the point at which the model achieves optimal performance on unseen data (on validation) and prevents it from overfitting. In addition to improving generalisation, early stopping conserves computational resources by eliminating unnecessary training steps.

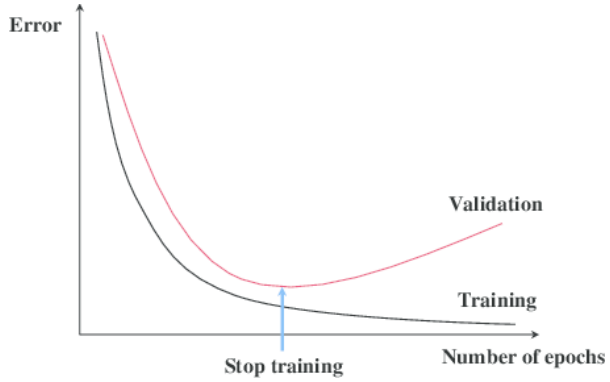


Figure 2.13: Illustration of early stopping. As training progresses, the training error (black curve) continues to decrease, while the validation error (red curve) reaches a minimum before increasing again, indicating overfitting. Early stopping halts training at the point where validation error starts to rise, preserving the model's generalisation ability.

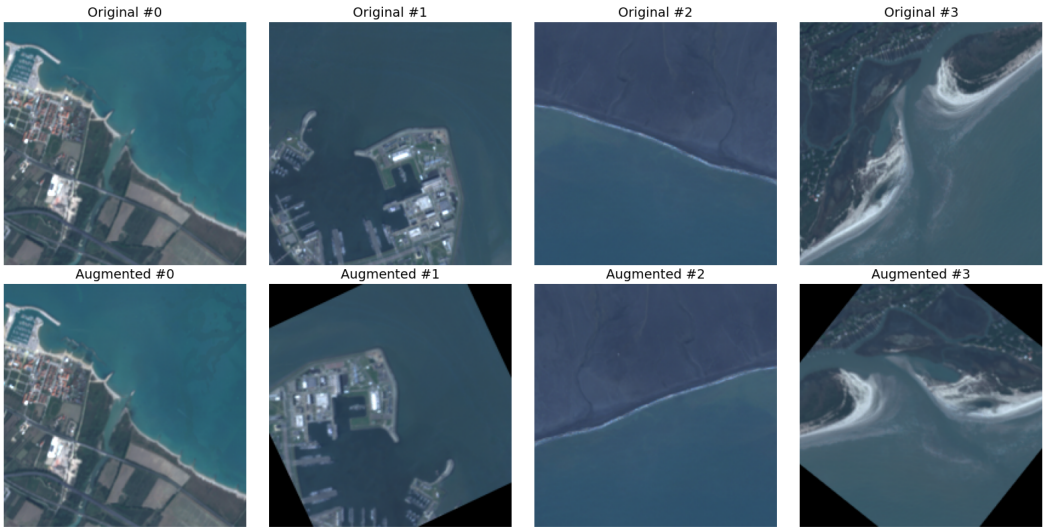


Figure 2.14: First 4 samples of a training batch. The top row shows original RGB satellite images; the bottom row shows corresponding augmented versions. Data augmentation includes random vertical flip, horizontal flip, and rotation. Augmentations are applied randomly, it is also possible that no augmentation is applied, as shown in image $\#0$.

Feature scaling and normalisation

Neural networks are sensitive to the scale of input features. To ensure stable and efficient learning, input features are typically scaled to a similar range. In this work, a Robust Min Max Scaler method was used: Robust, since it mitigates the impact of outliers by scaling data using the interquartile range,

which makes it fit to extreme values. Min Max ensures that the data is transformed to a fixed scale of $[0, 1]$.

Data augmentation

Augmentation introduces small, random variations to the training data, such as flipping, rotating, or scaling images. This encourages the model to learn more generalizable patterns rather than memorising specific examples. In this study, random horizontal and vertical flips and rotations were applied to satellite images to simulate different coastal orientations to improve training diversity and generalisation [44]. Figure 2.14 shows some examples of coastal training images before and after applying augmentations.

Hyperparameter Tuning

Hyperparameters are configuration values that are not learned from data, but manually set before training, such as batch size, learning rate, number of epochs, and model architecture choices. They significantly influence model performance and must be chosen carefully. Rather than picking hyperparameters arbitrarily, a search strategy is employed to explore different combinations. In this study, a Bayesian optimisation search was used to tune parameters of our CNN baseline, such as encoder type, learning rate, augmentation probability, and number of epochs. The goal was to find configurations that maximise the F1 score, finding the best configuration that performs best on unseen data.

Post-processing of predictions

After the model makes probabilistic predictions (between 0 and 1) for each type of class, these values are mapped to discrete classes (coastal type and shore type, for example). In this study, the maximum value per class was used to convert probabilities into binary decisions. This post-processing step is crucial in multi-label problems, where multiple thresholds may need tuning depending on the class distribution.

2.5. Prompting Techniques for LLMs

Prompting is the process of designing and structuring inputs to guide LLMs or MLLMs toward producing relevant and accurate outputs [45]. There exist plenty of different effective prompting strategies, and this section outlines several widely used prompting techniques relevant to our work, particularly in the context of coastal classification tasks.

2.5.1. Zero-shot prompting

In zero-shot learning, a model is asked to perform a task without being shown any labelled examples during the prompt. Instead, the model relies entirely on its pre-trained knowledge to infer the correct response. This approach is useful when labelled data is scarce or when evaluating generalisation capabilities.

2.5.2. Few-shot prompting

Few-shot learning, on the other hand, includes a small number of labelled examples within the prompt. These examples serve as guidance and context, helping the model better understand the structure of the task. Few-shot prompting can significantly improve performance, especially when language models are scaled up to have more parameters [46]. Analyzing which examples in the few-shot prompt are most effective in improving performance naturally leads to the problem of sample selection.

Sample Selection for Few-shot prompting.

The performance of few-shot learning can vary widely depending on the choice of examples included in the prompt. Effective sample selection strategies aim to cover diverse cases, reduce class imbalance, and provide clear contrasts between similar labels. In this study, we also experiment with different selection methods, including random sampling, and smarter sampling strategies to maximise the diversity and reduce the imbalance in the few-shot sample set to identify configurations that yield the best classification performance.

2.5.3. Chain-of-Thought Prompting

Chain-of-thought (CoT) prompting encourages the model to explain its reasoning before arriving at a final answer [47]. This approach is particularly effective in tasks that require multiple steps of reasoning or domain-specific knowledge. Zero-shot CoT essentially involves adding ”Let’s think step by step” to the original prompt to invoke the reasoning before answering the question [48].

2.5.4. Structured Output Formatting

To ensure consistency and facilitate automatic evaluation, prompts can be designed to elicit structured outputs. For example, requesting answers in a fixed format like JSON or bullet-point lists. Structured output formatting helps prevent ambiguous responses and supports reliable parsing of model predictions. In this study, we prompt the model to always answer with ”A1: <answer>, A2: answer”, so we are able to consistently parse and evaluate the responses of the MLLM.

3

Scientific Paper

Investigating Coastal Classification with Multi-Modal Large Language Models

Hugo de Heer
Computer Vision Lab - TU Delft
The Netherlands
h.j.deheer@student.tudelft.nl

Floris Calkoen
Deltares
The Netherlands
floris.calkoen@deltares.nl

Antonio Moreno-Rodenas
Deltares
The Netherlands
antonio.morenorodenas@deltares.nl

Jan van Gemert
Computer Vision Lab - TU Delft
The Netherlands
J.C.vanGemert@tudelft.nl

Abstract

Coastal zones are dynamic and vulnerable regions, demanding accurate, scalable monitoring tools to inform environmental management and hazard mitigation. While satellite imagery and CNN-based classifiers have improved automated mapping, their reliance on unstructured pixel data limits contextual understanding. This study presents the first fine-tuning of a multi-modal large language model (MLLM), Qwen2.5, on 12-channel satellite input for multi-label coastal classification, demonstrating how architectural adaptation enables integration of spectral, topographic, and derived features beyond RGB. We compare this approach to a ResNet-50 baseline and state-of-the-art prompting methods using GPT-4o and LLaMA-3.2. Our experiments on the CoastBench dataset reveal that MLLMs benefit substantially from few-shot prompting with diverse, balanced sampling and that fine-tuning Qwen2.5 with full 12-channel input outperforms its RGB-only variant. An ablation study quantifies the importance of elevation and water-sensitive indices, while a human benchmark exposes a performance ceiling near $F_1 \approx 0.70$ due to label ambiguity. Our findings suggest that while MLLMs can rival traditional models and offer interpretability benefits, future gains depend on dataset quality, input diversity, and prompting strategy design.

1. Introduction

The coastal zone is a dynamic interface where land meets the ocean; an area that encompasses diverse ecosystems such as beaches, dunes, estuaries, and deltas. It plays a critical role in the biosphere, known for its high biomass production, the provision of key ecosystem services, and its function as the first line of defence against coastal flooding [12]. The area is continually evolving due to natural factors, including wave action, tidal fluctuations and wind patterns, which contribute to its ever-changing landscape [30].

However, coastal zones are increasingly under pressure from both natural and anthropogenic factors. The frequency and intensity of coastal flooding events and the rate of coastal erosion are on the rise due to the changing climate [32], therefore posing a significant threat to coastal communities [24]. The rate of sea-level rise itself has been accelerating each year over the past century [9]. Projections indicate that without adaptation, a 0.75-meter rise in sea level would double the present at-risk population [16]. Altogether, this requires a timely and adequate commitment to adaptation.

Informed decision-making, which is urgently required for maintaining its habitability and essential ecosystem services, relies on robust data. Classifying coasts and shores into clear, distinct types can assist in monitoring environmental changes, disaster risk management and the preservation of ecosystems [2, 11, 15]. Although historical satellite catalogues and cloud computing platforms in recent years provided an unprecedented view of the coast [37], the potential of AI in coastal science remains largely untapped.

Deep learning techniques, such as convolutional neural networks (CNNs), have demonstrated remarkable success in image classification tasks [20, 31]. However, these models often lack interpretability and dynamic, high-level reasoning. Multi-modal large language models (MLLMs) offer a promising alternative by integrating visual and textual data, providing richer contextual understanding. This article investigates coastal classification using MLLMs by integrating satellite imagery with additional contextual information. To analyse to what extent MLLMs can enhance coastal classification using remote sensing data, we conduct experiments on a novel custom-developed coastal dataset

[8] and aim to answer the following sub-research questions:

- **RQ1:** How does a traditional CNN approach compare to untrained MLLMs in coastal classification?
- **RQ2:** What is the impact of different prompting techniques on the classification performance of MLLMs?
- **RQ3:** How can additional satellite bands, beyond RGB, be effectively embedded into an MLLM for coastal classification?
- **RQ4:** Which features contribute most to the performance differences observed across models and approaches?

The outline of the paper is as follows. Firstly, we introduce the coastal dataset in chapter 3, followed by chapter 4 containing the CNN baseline implementation including the experimental setup and results. After that, different various techniques are compared on pre-trained state-of-the-art (SOTA) MLLMs in chapter 5. Additional satellite bands, next to RGB, such as Near-Infrared (NIR) and height maps contain useful information for coastal classification, therefore chapter 6 investigates approaches on how to embed these into MLLMs. To answer RQ4, a feature ablation study is performed to identify key features in chapter 7. Finally, a discussion, conclusion and a future outlook are given in chapter 8 and chapter 9 respectively.

2. Related work

Coastal classification has traditionally relied upon geomorphologists and coastal scientists doing fieldwork [10, 29]. Recently, remote sensing techniques combined with machine learning models have been adopted. Hulskamp et al. (2023) [18] combined publicly available satellite imagery and coastal geospatial datasets, to train an automated classification method to identify muddy coasts with the use of a multispectral pixel-based random forest classifier. Another notable work in this area is by Dang et al. (2020) [13], which employed a CNN-based approach to classify coastal environments in Vietnam. Their methodology involved extracting 2D features along coastal transects, using geospatial attributes such as Digital Elevation Model (DEM), relative elevation, flow length, and slope. While effective within the targeted region, this approach may struggle with generalization to diverse coastal environments due to its reliance on region-specific features. Furthermore, the classification was limited to regional coastal types without incorporating important additional dimensions such as shoreline composition, the presence of buildings and or coastal defences.

Recent advances in MLLMs have introduced new ways to interpret and analyze remote sensing data. Kuchreja et al. (2023) [19] introduced GeoChat, a grounded MLLM designed for remote sensing applications. This model enables functionalities such as image captioning, visual question answering (VQA), and multi-turn conversations. However, while powerful, GeoChat was not specifically designed for coastal classification and lacks domain-specific data necessary for accurate coastal environment interpretation.

Similarly, SkyScript by Wang et al. [34] presents a large, semantically diverse dataset that maps geocoordinates to OpenStreetMap (OSM) image-text pairs, providing a valuable resource for remote sensing applications. However, the dataset does not contain specific coastal classifications, limiting its utility in tasks that require detailed coastal type differentiation. The absence of crucial coastal categories in both GeoChat and SkyScript highlights a gap in the application of MLLMs to coastal classification.

To the best of our knowledge, this study is the first to investigate the application of MLLMs to coastal classification. We use CoastBench [8], a custom coastal classification dataset specifically designed to better understand historical shoreline response and to map areas that are prone to coastal erosion and flooding. By leveraging MLLMs, we aim to evaluate not only their classification performance and interpretability, but also their capacity to be adapted for domain-specific coastal tasks. This is a key step towards deploying general AI systems for specialised environmental applications.

3. CoastBench Dataset

The dataset used in this study consists of coastal imagery and corresponding multi-label annotations, specifically designed to support the classification of coastal and shore types.

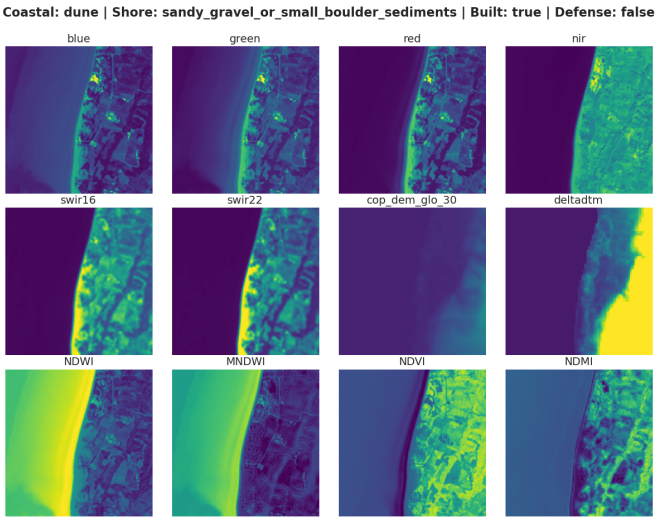


Figure 1. Example image containing all 12 channels of a coastal transect in CoastBench with a *Dune* coastline, *Sandy Gravel* shore type and containing a built environment with no coastal defences.

The dataset [8] is actively being developed and will be publicly released with (Calkoen 2025, A novel 100-m global coastal typology, in progress).

The CoastBench version (2025-01-27) we used currently covers 1090 coastal transects across Europe, with the number of transects continuing to grow. Each coastal transect covers a square area of approximately 8 km^2. The dataset contains a wide variety of coastal environments. Figure 1 shows an example of a coastline located in Greece, near the Ionian Sea.

Each image is a 200 x 200 pixel patch, patch, at 10 m spatial resolution. This image chip size balances the trade-off between image quality and computational efficiency, allowing for detailed coastal feature extraction while maintaining manageable computational requirements for future experiments below. The imagery is retrieved from Sentinel-2 satellite observations, with additional elevation data from the Copernicus Digital Elevation Model (DEM) [14] and Deltares Digital Terrain Model (DeltaDTM) [27]. The dataset employs median composites from the 10 least-clouded Sentinel-2 images in 2023, enhancing feature consistency and minimizing noise.

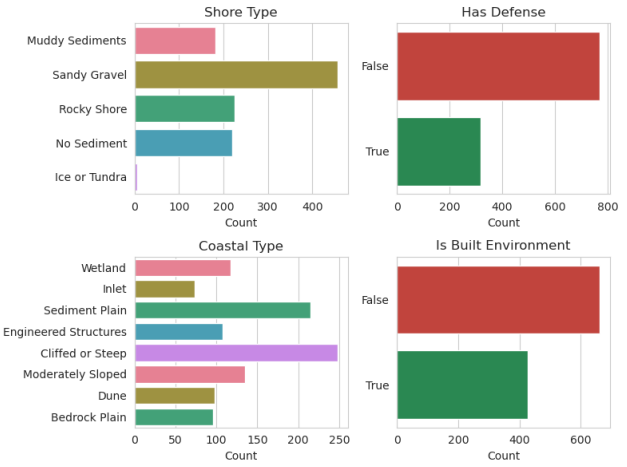


Figure 2. Label distribution of the CoastBench dataset.

The dataset includes 12 spectral and topographic channels and multi-label annotations for coastal and shore types, as well as the built environment and defence presence. A detailed specification of the channels and label classes is provided in Appendix sections A.1, A.2 respectively. We can see that the dataset is unbalanced by observing the label distribution in Figure 2. The multi-label structure reflects the complexity of coastal environments, where multiple characteristics can coexist within a single transect. This dataset serves as the foundation for performing the experiments in this work.

4. CNN-Based Coastal Classification Baseline

To establish an initial baseline for future comparison against other MLLM models, we introduce a novel CNN-based solution. Given a dataset image, it is the task of the CNN to classify the correct coastal type (c), shore type (s), and to predict visibility of built environment (b) and coastal defences (d).

4.1. Experimental setup

The CNN-based baseline model was trained on all 12 channels for multi-label coastal classification, with four one-hot encoded label sets (c, s, b, and d) concatenated into a single output vector.

All feature layers were normalized using a robust MinMaxScaler, scaling between the minimum and 99.5th percentile to mitigate high outliers. Class imbalance was addressed using a weighted out-of-fold F_1 score, ensuring that all classes were evaluated fairly despite differences in frequency. F_1 is also used as the main metric throughout this research.

A 5-fold group cross-validation was applied, grouping by coastal transect identifier to prevent leakage of the same area appearing in train and validation. Random rotations and flips were used as augmentations to improve generalization.

The model was trained using `BCELoss` [28] with a Sigmoid activation function. Hyperparameters, including encoder type, batch size, augmentation probabilities, learning rate, and epochs, were optimized via a Bayes-optimized grid search using Wandb Sweeps [5]. These results are listed in detail in Appendix section B.

Experiments were conducted on an NVIDIA RTX 3090 GPU with 128 GB RAM and an AMD Ryzen 9 7900 CPU running Ubuntu 22, averaging $\sim 1h$ per run.

4.2. Results

Table 1 shows the F_1 scores of the run *resnet-50* with the configuration that performed best in the hyperparameter sweep. An RGB run (*resnet-50-rgb*) is also included to compare later on with MLLM models, which normally only take in RGB channels. The chosen hyperparameter configuration can be seen in Table 8.

Run Type	$F_1(c)$	$F_1(s)$	$F_1(b)$	$F_1(d)$	Avg F_1
resnet50	0.578	0.700	0.860	0.825	**0.720**
resnet50-rgb	0.465	0.653	0.861	0.784	**0.662**

Table 1. Performance of the CNN-based coastal classification model, reporting F_1 scores for coastal type (c), shore type (s), built environment visibility (b), and coastal defences (d). Results are shown for the best-performing configurations.

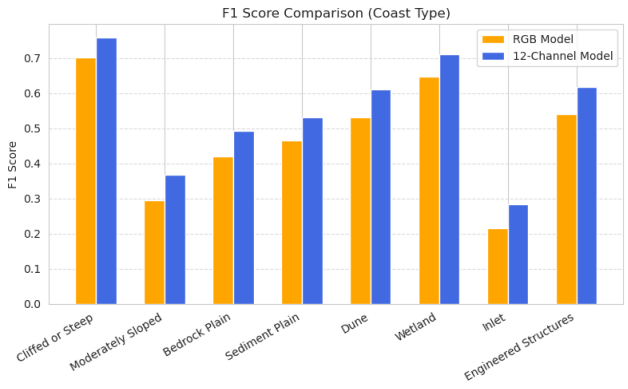


Figure 3. $F_1(c)$ comparison of all coastal classes between Run Type *resnet50* and *resnet50-rgb*.

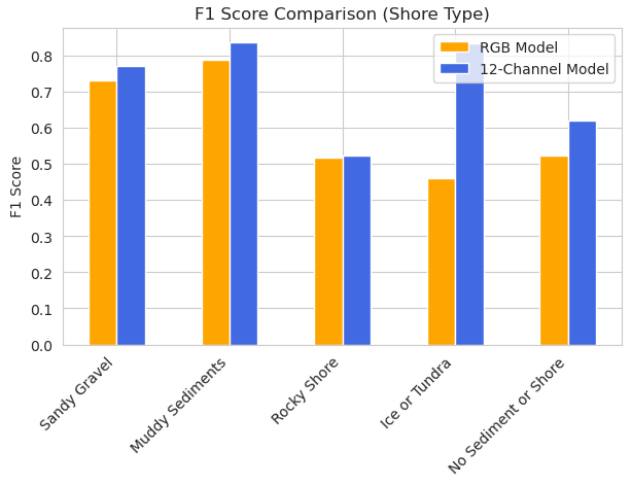


Figure 4. $F_1(s)$ comparison of all shore classes between Run Type *resnet50* and *resnet50-rgb*.

From the confusion matrices in Figures 15a and 15b, we observe strong classification performance for the *Cliffed or Steep* and *Dune* coastal types. The *Wetland* class is also well-identified, though some misclassification occurs with *Sediment Plain*, likely due to shared terrain characteristics. More pronounced confusion is observed in underrepresented categories such as *Inlet* and *Moderately Sloped*, which exhibit lower F_1 scores and a higher degree of misclassification. Notably, Figure 3 highlights that the 12-channel model consistently outperforms the RGB-only model across all coastal types, reinforcing the added value of multispectral and topographic information in coastal classification.

For shore type classification (Figures 16a and 16b), *Muddy Sediments* emerges as the best-classified category, with high precision and minimal confusion. In contrast, a notable three-way confusion is evident among *No Sediment*, *Rocky Shore*, and *Sandy Gravel*, suggesting textural overlap between these shore types. Interestingly, Figure 4 reveals that the 12-channel model generally improves classification performance for most shore types, except for *Rocky Shore*, where additional spectral bands did not contribute to better discrimination.

For built environment detection, the performance of the RGB-only model is nearly identical to that of the 12-channel model, indicating that additional spectral data does not significantly enhance classification in this category. Conversely, a clear improvement is observed for coastal defence detection, where the 12-channel model demonstrates a higher $F_1(d)$ score. This suggests that non-RGB spectral bands contribute valuable information for identifying coastal defence structures, potentially by capturing differences in material composition and elevation.

5. Examining Prompting Techniques in MLLMs

To address RQ2, we evaluate three prompting methods across three state-of-the-art MLLMs for coastal classification. Each model receives a system prompt outlining the task, a user prompt providing coastal-related questions, and an RGB image of a coastal transect. The models predict four attributes: coastal type (c), shore type (s), built environment visibility (b), and coastal defences (d).

5.1. Models

The selection of MLLMs for this study was guided by the need to balance performance, accessibility, and representational diversity in handling coastal classification tasks. We evaluated several candidate models and ultimately selected three state-of-the-art MLLMs; one commercial model and two smaller open-source models with manageable[1] fine-tune size. All original pretrained model weights were left unchanged for these experiments.

- **GPT-4o** - Developed by OpenAI and released in May 2024, this model is known for its strong multimodal capabilities [23] and robust reasoning abilities. ChatGPT-4o [23] is a leading commercial model, and also the largest model used over all of our experiments, demonstrating the State-of-the-art and making it a strong baseline for our coastal classification task.

- **Qwen2.5-VL-7B-Instruct** - A lightweight yet powerful open-source vision-language model developed by Alibaba released in February 2025. The leading Qwen2.5-VL-72B model matches state-of-the-art models like GPT-4o and Claude 3.5 Sonnet in multimodal tasks [4]. The smaller model of 7B parameters

[1]For fine-tuning experiments, we consider models with fewer than 15B parameters to be manageable on a single high-end GPU (e.g., an A100 80GB or RTX 4090 24GB) without requiring extensive multi-GPU setups.

is selected here, for efficient finetunability and to provide insight into the performance of more accessible, lower-resource models.

- **Llama-3.2-11B-Vision-Instruct** - Lastly, Meta's latest vision-language model released in September 2024, optimized for visual recognition, image reasoning and answering general questions about an image [3].

These models were chosen to ensure a mix of high-performance proprietary models (GPT-4o) and competitive open-source alternatives (Qwen2.5-VL-7B-Instruct and Llama-3.2-11B-Vision-Instruct).

5.2. Prompting Techniques

We compare three different prompting techniques presented in Table 2. System prompts as context (C), user prompts as questions (Q) and the coastal transects as images (I) were provided, where the model has to predict answer (A). All the specific prompts used can be seen in Appendix. C.1. To prepare the images, RGB channels were extracted from the dataset and a robust MinMaxScaler was applied.

Prompt	Method	Format
V_1	Structured Zero-shot	$CQI \rightarrow A$
$V_{1.1}$	V_1 with CoT	$CRQI \rightarrow A$
V_2	Structured Few-shot	$C(Q_iI_iA_i)_nQI \rightarrow A$

Table 2. Overview of the three prompting techniques used for coastal classification. V_1 represents structured zero-shot prompting, $V_{1.1}$ adds Chain-of-Thought reasoning, and V_2 incorporates a multi-modal few-shot approach.

V_1 is included as a baseline where we can analyse the zero-shot performance of the models. We provide the context C with question Q and image I, where the model produces a structured response answering c, s, b d.

Chain-of-Thought (CoT) prompting has been shown to increase performance and elicits reasoning in large language models [35], [36]. Therefore we introduce $V_{1.1}$ with additional reasoning (R), to generate intermediate reasoning chains to infer the answer.

Providing examples can be used to enable in-context learning to steer the model to have a more consistent and performant output [7,33]. V_2 consists of a multi-modal few-shot approach, where n examples are given with coastal questions Q, images I and answers A as context. n was chosen to be 10, after experimentation with GPT-4o, giving the best performance.

5.3. Few-shot selection strategy

In few-shot multimodal prompting, the choice of support examples critically impacts model performance. Unlike traditional supervised learning, where all training samples can be used, few-shot prompting is constrained by model input length and inference costs. In our case, we cannot provide the full dataset of 1100 coastal transects, but only a small subset fits within the model's context. Prior work in 2024 compared few-shot sampling techniques, but this was limited to textual LLMs [26].

In this study, it is hypothesised that selecting the optimal few-shot subset is crucial for maximising performance in our novel setting, where we apply MLLMs for multiclass and multi-label coastal classification. Prior work in 2024 compared few-shot sampling techniques, but this was limited to textual LLMs [26].

Our goal is to maximize **diversity** (informativeness) and **balance** to embed as much class information as possible while keeping the sample size low. This approach reduces costs and mitigates bias toward specific classes.

We define two metrics to guide our selection: diversity and balance.

5.3.1 Diversity

The diversity score (DIV) measures how many unique feature values are covered relative to the total number of possible values:

$$DIV = \frac{|C| + |S| + |B| + |D|}{|C^*| + |S^*| + |B^*| + |D^*|}$$

where:

- C, S, B, D are the sets of unique values for coastal type, shore type, built environment, and defence presence in the selected subset.
- C^*, S^*, B^*, D^* are the total possible unique values for each feature.

5.3.2 Balance

The balance score (BAL) is based on the entropy of the feature distribution:

$$BAL = \frac{H_C + H_S + H_B + H_D}{\log_2 |C^*| + \log_2 |S^*| + \log_2 |B^*| + \log_2 |D^*|}$$

where:

$$H_X = -\sum_i p_i \log_2 p_i$$

is the entropy for each feature X, and p_i is the probability of observing feature value i in the selected subset.

5.3.3 Selection Algorithms

Four distinct selection strategies are implemented to efficiently sample diverse and balanced few-shot subsets:

- **Random**: A baseline approach where we randomly select n samples without considering diversity or balance.
- **Brute (brute-force diverse sampling)**: This method iteratively selects samples that introduce novel coastal and shore types while avoiding duplicates. When diversity cannot be further increased, it resorts to random selection to complete the subset.
- **Greedy (greedy diverse sampling)**: For each iteration, this strategy evaluates all remaining samples and selects the one that maximises the number of unseen feature values across coastal type, shore type, built environment, and defence presence.
- **Greedy Balanced (greedy balanced diverse sampling)**: An extension of the greedy approach that not only prioritises new feature values but also balances the selection by favouring underrepresented feature values using inverse frequency weighting. Pseudocode for this method is listed in Listing 5.

5.3.4 Few-shot algorithm results

Diversity and balance scores of all four algorithms were tested over 10 different random seed runs. Results shown in Figure 5 demonstrate that the *random* selection method performs the worst in both diversity and balance metrics, which is expected due to its lack of strategic selection. Furthermore, the *brute-force* approach shows improvement in diversity but struggles to achieve balance as efficiently. The *greedy* algorithm outperforms the brute-force method in both diversity and balance, especially in the early few-shot numbers. The *greedy-balanced* approach achieves the highest balance score, consistently outperforming other methods, while also reaching near-optimal diversity. These results confirm that explicitly accounting for underrepresented feature values (*greedy-balanced*) leads to more balanced and diverse few-shot sample selection. Section 5.4 will present the model results and also look at the correlation between the balance, diversity and model F_1 scores.

5.4. Model results

Figure 6 presents the F_1 scores for all models across the three prompting versions: V_1 (zero-shot), $V_{1.1}$ (zero-shot with reasoning), and V_2 (few-shot). A table with all scores can be seen in Table 9.

Overall, *gpt-4o* achieves the highest performance across all categories and benefits significantly from few-shot prompting, as evidenced by the sharp increase in scores from V_1 to V_2. Notably, it outperforms the trained *resnet50-rgb* baseline, reaching a maximum F_1 score of 0.689 compared to the CNN's 0.662.

Qwen2.5 follows closely behind, with substantial improvements in F_b (built environment) and F_d (defence) when using few-shot prompting. However, it struggles more with coastal and shore classifications, often overpredicting classes found in the first few-shot example. Interestingly, when reasoning is introduced in zero-shot ($V_{1.1}$), *qwen2.5* experiences a drop in F_b and F_d, frequently hallucinating and overpredicting those classes.

Llama-3.2 demonstrates competitive zero-shot performance relative to *qwen2.5* but struggles significantly in few-shot settings. It tends to predict a limited subset of classes,

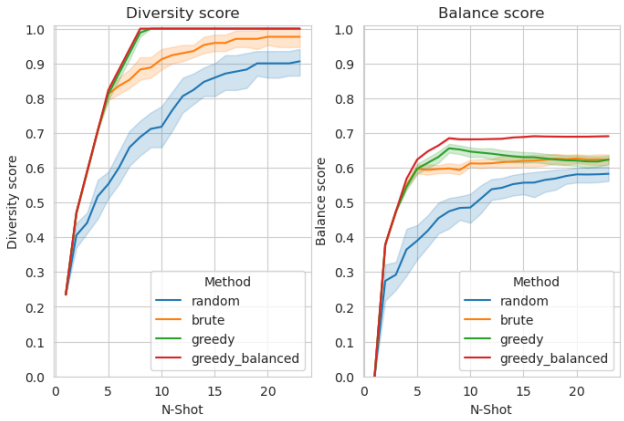


Figure 5. Diversity (left) and balance (right) scores of four different few-shot selection algorithms as a function of the number of shots (N-Shot). The solid lines represent the mean diversity and balance scores across different trials, while the shaded regions indicate the variability (95% confidence interval).

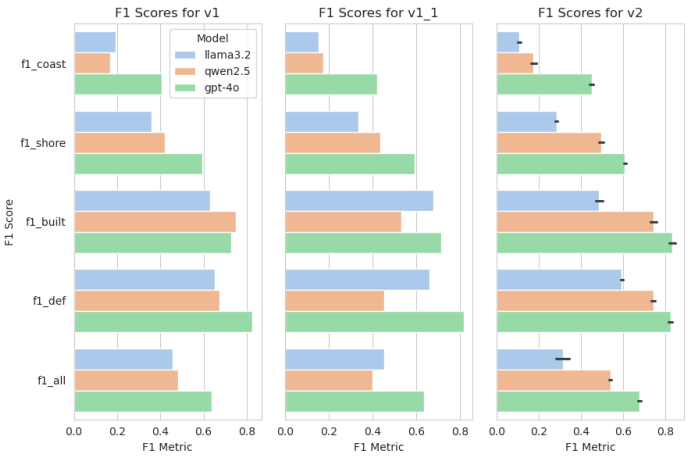


Figure 6. F_1 scores for each model across different prompting strategies (V_1, $V_{1.1}$, and V_2)

heavily favoring specific categories such as *Cliffed or Steep and Moderately Sloped*, even when these are not overrepresented in the few-shot subset.

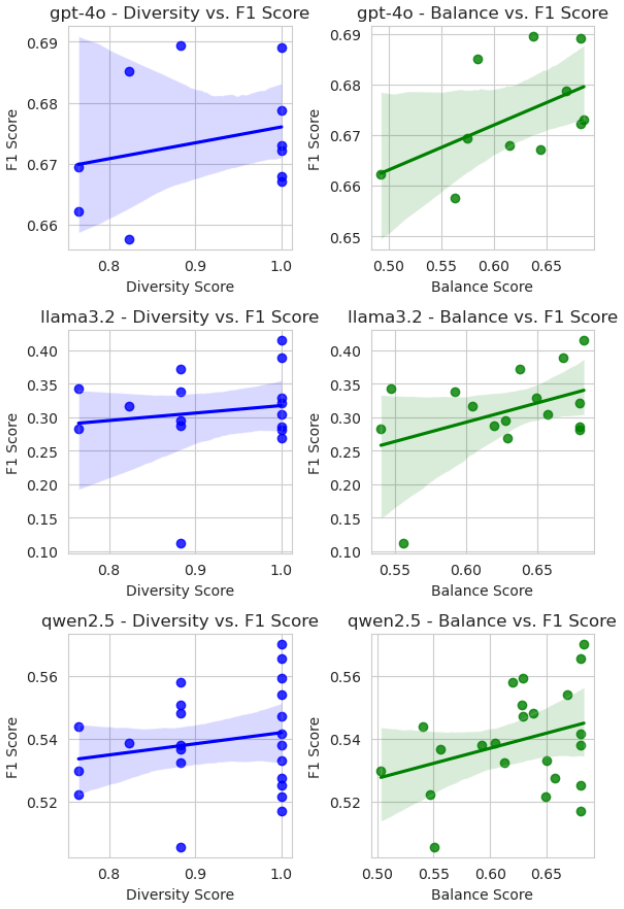


Figure 7. Relationship between diversity, balance, and F_1 scores for different few-shot selection methods. Each subplot shows the correlation between the diversity score (left column) or balance score (right column) with F_1 scores for GPT-4o, LLaMA 3.2, and Qwen 2.5. The solid lines indicate the mean trend, while the shaded regions represent the 95% confidence interval (CI), showing the uncertainty in the regression estimates.

In Figure 7, we can find the relationship between diversity, balance, and F_1 scores for different few-shot selection methods. We observe a positive correlation between these scores and their corresponding F_1 scores for all models. This indicates that a higher diversity and balance in the few-shot selection subset contribute to improved performance.

Among the different categories observed in Figure 19, we note that the coastal (c) and shore (s) classes show a particularly strong improvement with increased diversity and balance, especially for Qwen-2.5. This suggests that these categories benefit more from a varied selection of examples, due to being multi-label. In contrast, for GPT-4o, the improvement is less pronounced, indicating that the model performs well across different diversity and balance levels. For the built (b) and defence (d) categories, the impact of diversity and balance appears more neutral, with GPT-4o maintaining high performance regardless of variation. LLaMA-3.2, however, continues to struggle across all categories, reinforcing the observation that its overall performance is more constrained by model limitations rather than prompt selection strategies.

6. Embedding of Additional Satellite Bands and Feature Maps into MLLMs

MLLMs have typically been pretrained on standard RGB imagery, whereas satellite data has increased spectral depth. It is common to have 10 bands in satellite data, while newer hyperspectral satellite sensors measure across 200+ bands [22, 25]. In this study, it is hypothesised that the performance of MLLMs will increase with the additional information coming from extra bands. This is addressed in RQ3: How can additional satellite bands, beyond RGB, be effectively embedded into an MLLM. Here, we experiment with extending the input of Qwen-2.5 to support multi-channel satellite imagery and evaluate its performance on the Coast-Bench dataset.

6.1. Experimental Setup

We focus exclusively on architectural modification through fine-tuning of the Qwen2.5-VL-7B-Instruct model. This model was selected because it is open-source, has a manageable fine-tuning size, and has a strong performance in earlier prompting experiments (see Section 5.4).

To enable ingestion of additional satellite bands and support multi-channel learning, the Qwen2.5 vision encoder and training pipeline were modified as follows:

- The first convolutional patch embedding layer of the vision encoder was replaced to accept n-channel inputs. For the standard RGB channels, pretrained weights were copied directly. The remaining channels (e.g., NIR, SWIR, DEM) were initialised by using He initialisation to facilitate efficient training [17].
- The Hugging Face implementation of Qwen2.5 was adapted to support non-standard input formats using `.npy` arrays. This included modifications to the vision tokeniser, saving and loading the model, and input transforms.
- The original Qwen2-VL-Finetune script was extended to enable: (i) Grouped 5-fold cross-validation using the same splits and augmentations as the CNN baseline. (ii) Custom validation set to support multi-label classification metrics across four outputs: coastal type, shore type, built environment, and coastal defence.

We evaluated the model under two input variants:

1. *Qwen2.5*: Full multispectral and topographic input as defined in Table 6.

2. *Qwen2.5-rgb*: RGB only (fine-tuned MLLM baseline),

To reduce overfitting and preserve generalisation capabilities, we applied LoRA (Low-Rank Adaptation) to the vision encoder. This allows efficient fine-tuning while freezing the language decoder and other pretrained components, mitigating catastrophic forgetting [21] on our relatively small CoastBench dataset.

All experiments were conducted on a single NVIDIA H100 GPU, with each cross-validation fold averaging approximately 1 hour. All training parameters (optimiser, scheduler, batch size) were kept at the recommended defaults of the original Qwen2.5 fine-tuning framework. Full training configuration details are provided in Appendix Table 10.

6.2. Model results

Table 3 shows the F_1 out-of-fold scores of the run *qwen2.5* fine-tuned on all 12 channels. (*qwen2.5-rgb*) presents the training run that has been trained on 3 channels (RGB).

Run Type	$F_1(c)$	$F_1(s)$	$F_1(b)$	$F_1(d)$	Avg F_1
qwen2.5	0.556	0.659	0.789	0.793	**0.700**
qwen2.5-rgb	0.503	0.618	0.813	0.811	**0.686**

Table 3. Performance of the Qwen2.5-based fine-tuned coastal classification models, reporting F_1 scores for coastal type (c), shore type (s), built environment visibility (b), and coastal defences (d). Results are shown for the best-performing configurations.

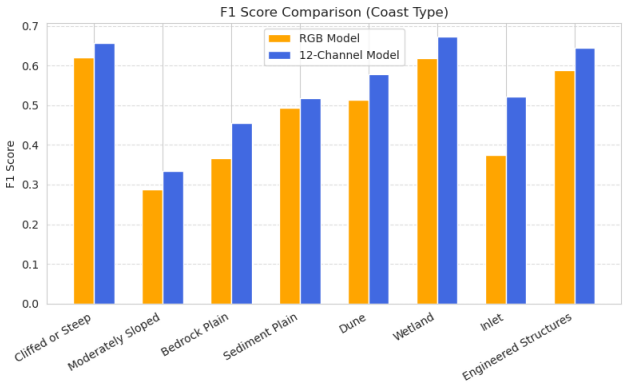


Figure 8. $F_1(c)$ comparison of all coastal classes between Run Type *qwen2.5* and *qwen2.5-rgb*.

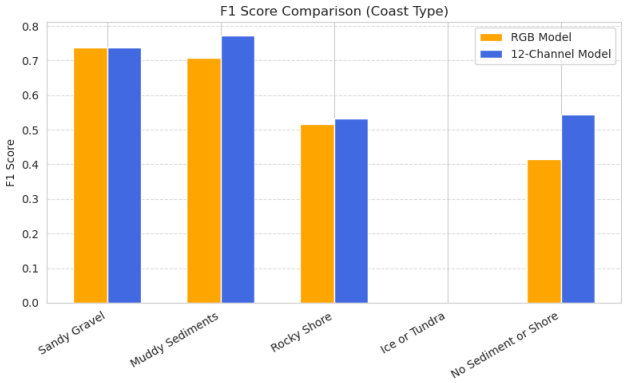


Figure 9. $F_1(s)$ comparison of all coastal classes between Run Type *qwen2.5* and *qwen2.5-rgb*.

Table 3 and Figures 20 and 21 show that incorporating nine additional spectral and topographic channels yields consistent improvements in both coastal and shore classification. In the eight-class coastal task (Figure 20), *Cliffed or Steep* and *Wetland* remain the best-classified categories, with modest F_1 gains of +0.03–0.04. The most significant improvement occurs for *Inlet*, which jumps by +0.18, reflecting enhanced delineation of narrow water inlets. *Wetland* and *Sediment Plain* also rise by +0.04–0.05, whereas underrepresented classes such as *Moderately Sloped* and *Bedrock Plain* see only marginal gains. Notably, across all eight coastal types the 12-channel model outperforms the RGB-only variant, underscoring the value of multispectral and elevation cues.

Similarly, for the five-class shore task (Figure 21), *Muddy Sediments* continues to be the top-performer, and confusion persists among *No Sediment or Shore Platform*, *Rocky Shore*, and *Sandy Gravel* due to textural overlap. Yet every shore class benefits from additional channels: *No Sediment or Shore Platform* improves by +0.10, *Inlet* by +0.08, and even *Rocky Shore* gains +0.02, highlighting how spectral diversity aids in making distinctions between sediments.

In contrast, the binary detection tasks reveal slight performance declines (Figure 22). *Built Environment* F_1 falls from 0.813 to 0.789, and *Coastal defence* drops from 0.811 to 0.793. This suggests that while RGB imagery sufficiently captures high-contrast human structures, the additional channels may introduce noise or divert model capacity away from anthropogenic feature recognition.

7. Feature Ablation

The goal of the ablation study is to quantify the partial contribution of each input channel (and coherent groups of channels) to the classification performance of our best CNN model, denoted *cnn-best*.

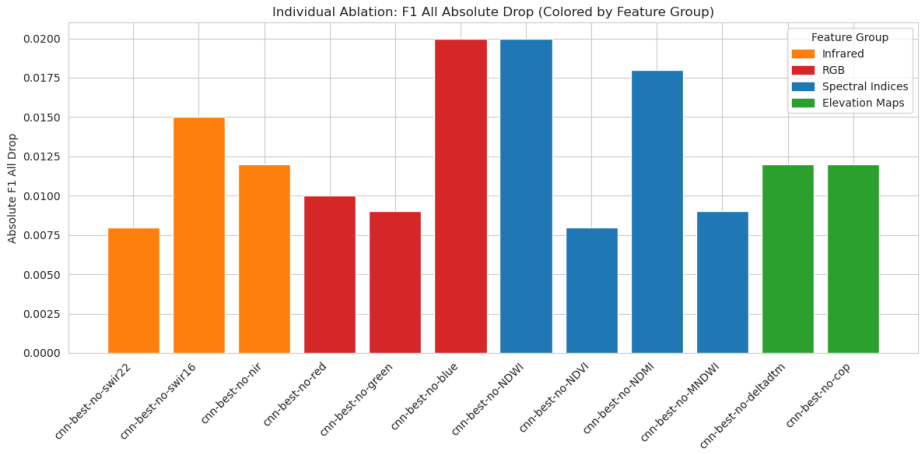


Figure 10. Individual ablation: mean absolute drop in overall F1 per removed channel, colored by feature group.

7.1. Experimental Setup

Two tiers of experiments are defined to structure the ablation analysis. In the *grouped ablation*, one entire feature group is withheld at a time, specifically:

1. **Infrared bands:** (NIR, SWIR-16, SWIR-26)
2. **Human-visible bands**: (Red, Green, Blue)
3. **Spectral indices**: (NDWI, MNDWI, NDVI, NDMI)
4. **Elevation maps**: (COP DEM GLO and DeltaDTM)

In the *individual ablation*, twelve separate models are retrained, each missing exactly one of the original input channels or indices.

All ablation runs are trained from scratch using the identical data splits, augmentation pipeline, and hyperparameter settings as the baseline. Performance is evaluated by the per-class F_1 score on *Coast*, *Shore*, *Built*, and *Defence*. For each ablation experiment, we report the *mean absolute drop* in F_1:

$$\Delta F_1 = \frac{1}{N}\sum_{i=1}^{N}\left(F_1^{(i)}(\text{baseline}) - F_1^{(i)}(\text{ablation})\right),$$

where $N = 10$ corresponds to the total number of runs (5-fold cross-validation repeated with 2 random seeds). The corresponding relative change in percent is computed as:

$$\%\Delta F_1 = 100 \times \frac{\Delta F_1}{\frac{1}{N}\sum_{i=1}^{N} F_1^{(i)}(\text{baseline})}.$$

To assess the statistical significance of observed performance changes, a two-sided paired t-test is applied to the average F_1 scores of the baseline and ablation across all $N = 10$ runs.

7.2. Results

Table 4 presents the grouped-ablation outcomes. The infrared causes the largest decrease in overall F_1 ($\Delta F_1 = 0.020$, 2.72 %). Most of the infrared ablation effect is driven by a 0.030 drop in the Shore F_1 score. By contrast, elevation information is crucial for accurately classifying coastal morphologies, such as Cliffed, Steep, and Moderately Sloped, because the terrain gradients provided by DEM and DeltaDTM enable increased differentiation between these classes (Table 12). The RGB bands are especially important for detecting built structures. When these bands are removed, the Built F_1 score decreases from 0.859 to 0.837. Spectral indices also contribute to detecting artificial structures such as coastal defences. Their removal results in a Defence F_1 drop from 0.824 to 0.807, indicating their utility in capturing spectral contrasts between wet and dry surfaces that are often associated with man-made structures. Figure 11 illustrates these absolute drops by feature group.

Ablation	$\boldsymbol{\Delta F_1}$	$\boldsymbol{\%\Delta F_1}$	p**-value**
No infrared	0.020 ± 0.017	2.72 %	0.007
No RGB	0.016 ± 0.023	2.16 %	0.070
No spectral indices	0.014 ± 0.028	1.98 %	0.149
No elevation maps	0.014 ± 0.016	2.02 %	0.026

Table 4. Grouped ablation: absolute F_1 ($\Delta F_1 \pm$ std) and relative ($\%\Delta F_1$) drops in overall F_1 along with p-values.

The individual-ablation experiments (Table 5 and Figure 10) show that removing Blue causes the largest overall performance drop ($\Delta F_1 = 0.020$, 2.74%). Furthermore, NDWI appears beneficial for detecting defence structures, where its removal leads to a decrease in the Defence F_1 score of $\Delta F_1(d) = 0.015$ (Table 11). This finding aligns

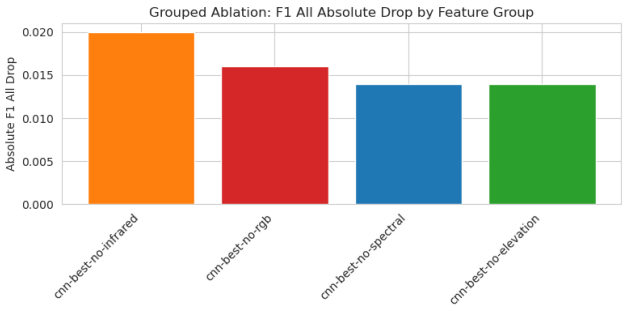


Figure 11. Grouped ablation: mean absolute drop in overall F_1 when each feature group is withheld.

Ablation	ΔF_1	$\%\Delta F_1$	p-value
No SWIR-26	0.008 ± 0.030	1.10 %	0.456
No SWIR-16	0.015 ± 0.025	2.11 %	0.104
No NIR	0.012 ± 0.031	1.71 %	0.260
No Red	0.010 ± 0.020	1.33 %	0.179
No Green	0.009 ± 0.019	1.30 %	0.179
No Blue	0.020 ± 0.014	2.74 %	0.002
No NDWI	0.020 ± 0.023	2.72 %	0.030
No NDVI	0.008 ± 0.022	1.16 %	0.288
No NDMI	0.018 ± 0.026	2.52 %	0.067
No MNDWI	0.009 ± 0.021	1.29 %	0.217
No DeltaDTM	0.012 ± 0.025	1.67 %	0.178
No COP DEM GLO	0.012 ± 0.031	1.61 %	0.292

Table 5. Individual ablation: absolute ($\Delta F_1 \pm$ std) and relative ($\%\Delta F_1$) drops in overall F_1, along with p-values.

with the grouped ablation results, indicating that NDWI helps the model distinguish defence features at the water's edge. In contrast, removing a single elevation input does not substantially affect F1 coast performance, unlike the grouped ablation, where both were removed together. This suggests that either elevation source alone provides sufficient topographic information for the model to differentiate between coastal morphologies.

8. Discussion

To assess the upper bound of coastal classification performance, a human benchmark was established. Two coastal experts from Deltares manually annotated 50 transects selected using a greedy balanced diverse sampling method. Unlike the models, these experts were allowed to freely explore the surrounding region, access contextual cues (e.g., surrounding morphology, infrastructure), and draw from domain knowledge. Despite this advantage, the resulting scores hovered around an average F1 of 0.68–0.70 across categories, marginally below the best-performing *resnet50* model (0.72) as seen in Figure 12.

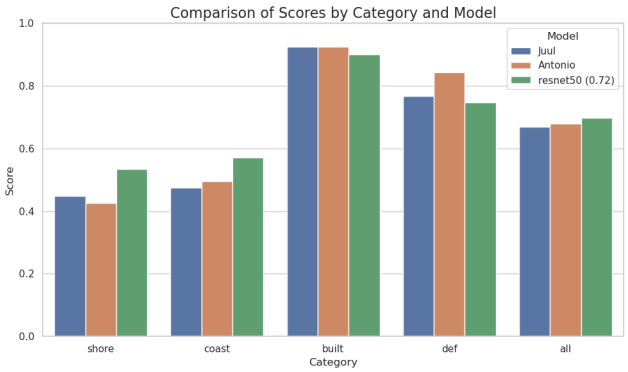


Figure 12. Comparison of model performance and human annotations on 50 diverse coastal transects. Two coastal experts labelled each transect independently using all available spatial context, while the *resnet50* model relied solely on static image input.

This outcome highlights two key insights. First, the *resnet50* baseline approaches the limits of what is currently achievable under this label regime. Second, and more importantly, it reveals underlying ambiguity in the labelling task itself. For example, experts often disagreed on the boundaries between dune and sediment plain, or between inlet and wetland,

In this light, model optimisation beyond 0.70 may be less a question of better architectures and more a matter of redefining the problem setup. Future work might benefit from embracing uncertainty through probabilistic labelling or incorporating multiple annotator perspectives to reflect multi-class coasts. Some examples of difficult-to-classify coasts can be found in Appendix figures 23, 24 and 25.

Weight initialisation has long been a topic in GeoAI [6]. During fine-tuning of Qwen2.5 on 12-channel input, we found that the initialisation of the newly added input weights in the patch embedding layer plays a critical role in model performance. Simply replicating the pre-trained RGB weights across the additional channels consistently led to poor convergence, with the model plateauing around an F1 score of 0.55, suggesting being stuck in a suboptimal local minimum. The performance degradation was particularly pronounced when elevation maps (COP DEM GLO and DeltaDTM) were added as input channels. Compared to the near-infrared bands, which share some structural similarity with RGB in terms of spatial gradients and intensity distributions, the elevation maps exhibit entirely different statistical properties. As a result, copying pre-trained RGB weights into these new input channels created a strong inductive bias misaligned with the elevation domain, leading to a larger drop in F1 score and impaired convergence. Channel-wise feature similarity should inform weight initialisation strategies when extending pre-trained visual encoders, which could be a new compelling future research

lead. Further investigation into other weight initialisation strategies, and hyperparameter tuning could unlock additional gains trained on geospatial data.

An interesting observation during interaction with the fine-tuned *Qwen2.5-RGB* model was its ability to provide structured and plausible responses to images outside of the training distribution, as illustrated in Figures 26 and 27. These examples, drawn from non-European coastal scenes and lacking ground truth labels, were not part of the fine-tuning dataset. Nevertheless, the model produced predictions that were not only consistent with the visual features in the images but also aligned with reasoning in the dataset, such as detecting a harbour as being part of engineered structures. It is plausible, though, that these images, or visually similar ones, were encountered during the model's original pretraining phase. In such cases, the model's predictions may reflect memorisation or prior exposure rather than true generalisation. During exploration, the model also showed to retain its general capabilities while being fine-tuned on this specific coastal task. However, interaction logs revealed limitations in the model's responsiveness to follow-up prompts. For instance, prompting the model for explainability by asking "please elaborate" sometimes led to repetition rather than deeper reasoning, unless more explicitly phrased (see Figure 27). This inconsistency suggests that fine-tuning MLLMs on small 1-turn datasets may demonstrate impressive zero-shot consistency on novel inputs, but they still struggle with interactive robustness and contextual understanding across turns. Further investigation and experimentation with the fine-tuned model would be necessary to validate these findings.

9. Conclusion and Future Work

This study presents the first systematic investigation into the application of Multi-Modal Large Language Models (MLLMs) for automated coastal classification using remote sensing data. By benchmarking traditional CNNs against zero-shot and fine-tuned MLLMs under varying prompting strategies and input modalities, we provide empirical evidence and conceptual insight into the potential of MLLMs for geospatial reasoning. The key findings of this work are:

- **Few-shot prompting improves performance:** MLLMs benefit significantly from well-curated, diverse, and balanced example sets, highlighting the importance of prompt composition.
- **Additional input channels are critical:** Extending input beyond RGB (e.g., NIR, SWIR, DEM, spectral indices) measurably improves classification outcomes.
- **Non-RGB integration requires architectural care:** Successfully embedding non-standard channels into the vision encoder of Qwen2.5 necessitates dedicated weight initialisation and tuning strategies.
- **Qwen2.5 can be reliably fine-tuned for structured outputs:** Fine-tuning Qwen2.5 on CoastBench enables the model to produce consistent and structured outputs tailored to the classification task.

Despite these advances, we found inconsistencies in the dataset's labelling confirmed through a human benchmark performed by two coastal experts, imposing an upper-bound on performance (F1 $\approx$ 0.70). These findings suggest that while MLLMs offer a scalable and interpretable alternative to conventional models, future progress may depend more on improving label quality, dataset size and contextual understanding than solely on architectural refinement.

9.1. Future Work

While this study demonstrates the feasibility of using MLLMs for coastal classification and represents an initial step toward the development of coastal AI agents, several directions remain open for further exploration:

- **Dataset quality and label reliability:** The observed performance ceiling (F1 $\approx$ 0.70) is likely influenced by hard-to-classify coasts and inconsistent labels. Future efforts should incorporate multi-expert voting or probabilistic labelling to better reflect the inherent uncertainty in coastal classification. Explicitly encoding the key area of interest (the coastal transect) may improve disambiguation in complex scenes.
- **Dataset scale:** The current dataset is limited to transects from European coastlines only. Expanding the dataset with globally diverse coastal morphologies would improve generalizability and expose model robustness to unseen environments.
- **Contextual prompting:** MLLM prompts could be enriched with metadata such as regional weather patterns, OpenStreetMap (OSM) tags, or local geomorphological descriptors to support more grounded predictions and reasoning. Meta prompting could also be explored as a structure-focused alternative, potentially improving performance and reducing token overhead.
- **Alternate input embedding strategies:** Instead of modifying the vision encoder, future work could investigate embedding additional satellite bands as grayscale auxiliary images directly in the prompt. Comparing this strategy with the fine-tuning approach may offer insights into flexibility versus performance trade-offs. Additionally, exploring advanced weight initialisation techniques and further hyperparameter optimisation for fine-tuned models like Qwen2.5 may yield better convergence.

- **Explainability assessment:** Future work could assess whether MLLMs can identify the features most responsible for explaining its coastal classification predictions. This would include exploring multi-turn interaction to enable MLLMs as coastal agents, not just classifiers.

References

[1] Abubakar Abid, Ali Chaudhary, Safwan Abdalla, Akshay Goyal, Sarthak Joshi, Kushal Misra, and James Zou. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*, 2019. 22

[2] K. Ahrendt, A. Scalise, H. Sterr, F. Müller, and I. Ruljevic. A new multifunctional coastal classification for ecosystem-service assessments. *Natural Resources Conservation and Research*, 1(1):1–10, 2018. 1

[3] Meta AI. Llama 3.2: Advancing vision ai for edge and mobile devices, 2024. Accessed: 2025-03-11. 5

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. 4

[5] Lukas Biewald. Experiment tracking with weights and biases. Software available from wandb.com, 2020. 3

[6] Wadii Boulila, Eman Alshanqiti, Ayyub Alzahem, Anis Koubaa, and Nabil Mlaiki. An effective weight initialization method for deep learning: Application to satellite image classification. *Expert Systems with Applications*, 254:124344, 2024. 10

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 5

[8] Floris Calkoen, Arjen Luijendijk, Antonio Moreno-Rodenas, and Stefan Aarninkhof. Mapping coastal typology using publicly available earth observation data and deep neural networks. *Coastal Engineering Proceedings*, page 158, 09 2023. 2, 3

[9] John A. Church and Neil J. White. A 20th century acceleration in global sea-level rise. *Geophysical Research Letters*, 33(1), 2006. 1

[10] European Commission. Living with coastal erosion in europe: Sediment and space for sustainability. Technical report, Directorate-General for Environment, Brussels, Belgium, 2004. 2

[11] J. A. G. Cooper and S. McLaughlin. Contemporary multidisciplinary approaches to coastal classification and environmental risk analysis. *Journal of Coastal Research*, 14(2):512–524, 1998. 1

[12] Christopher Crossland, Dan Baird, Jean-Paul Ducrotoy, Han Lindeboom, Robert Buddemeier, William Dennison, Bruce Maxwell, Stephen Smith, and Dennis Swaney. *The Coastal Zone — a Domain of Global Interactions*, pages 1–37. 03 2006. 1

[13] Kinh Bac Dang, Van Bao Dang, Quang Thanh Bui, Van Vuong Nguyen, Thi Phuong Nga Pham, and Van Liem Ngo. A convolutional neural network for coastal classification based on alos and noaa satellite data. *IEEE Access*, 8:11824–11839, 2020. 2

[14] European Space Agency. Copernicus global digital elevation model, 2024. Accessed: 2025-03-20. 3

[15] Charles Finkl. Coastal classification: Systematic approaches to consider in the development of a comprehensive scheme. *Journal of Coastal Research*, 20:166–213, 12 2004. 1

[16] Marjolijn Haasnoot, Gundula Winter, Sally Brown, Richard J. Dawson, Philip J. Ward, and Dirk Eilander. Long-term sea-level rise necessitates a commitment to adaptation: A first order assessment. *Climate Risk Management*, 34:100355, 2021. 1

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. 7

[18] Romy Hulskamp, Arjen Luijendijk, D.s Maren, Antonio Moreno-Rodenas, Floris Calkoen, Etienne Kras, Stef Lhermitte, and Stefan Aarninkhof. Global distribution and dynamics of muddy coasts. *Nature communications*, 14:8259, 12 2023. 2

[19] Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad Shahbaz Khan. Geochat: Grounded large vision-language model for remote sensing, 2023. 2

[20] Ying Li, Haokui Zhang, Xizhe Xue, Yenan Jiang, and Qiang Shen. Deep learning for remote sensing image classification: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(6):e1264, 2018. 1

[21] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025. 8

[22] Jürgen Nieke, Laurent Despoisse, Alberto Gabriele, Hans Weber, Hannes Strese, and Ferran Gascon. The copernicus hyperspectral imaging mission for the environment (chime): An overview of its mission, system and planning status. In *Proceedings of SPIE*, volume 12729, page 1272909. SPIE, 2023. 7

[23] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, and Rosie Campbell. Gpt-4 technical report, 2024. 4

[24] Tianze Pang, Xiuquan (Xander) Wang, Rana Nawaz, Genevieve Keefe, and Toyin Adekanmbi. Coastal erosion and climate change: A review on coastal-change process and modeling. *Ambio*, 52, 07 2023. 1

[25] Jay S Pearlman, Peter S Barry, Clifford C Segal, Jason Shepanski, Daniel Beiso, and Scott L Carman. Hyperion, a space-based imaging spectrometer. *IEEE Transactions on Geoscience and Remote Sensing*, 41(6):1160–1173, 2003. 7

[26] Branislav Pecher, Ivan Srba, Maria Bielikova, and Joaquin Vanschoren. Automatic combination of sample selection strategies for few-shot learning, 2024. 5

[27] Maarten Pronk, Aljosja Hooijer, Dirk Eilander, Arjen Haag, Tjalling de Jong, Michalis Vousdoukas, Ronald Vernimmen, Hugo Ledoux, and Marieke Eleveld. Deltadtm: A global coastal digital terrain model. *Scientific Data*, 11, 2024. 3

[28] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. 3

[29] Christopher Sharples. The Smartline - an effective coastal data mapping format. 1 2008. 2

[30] Andrew Short. Coastal processes and beaches. *Nat. Educ. Knowl.*, 3, 01 2012. 1

[31] Adam Stewart, Caleb Robinson, Isaac Corley, Anthony Ortiz, Juan Lavista Ferres, and Arindam Banerjee. Torchgeo: Deep learning with geospatial data. *ACM Transactions on Spatial Algorithms and Systems*, 12 2024. 1

[32] Mohsen Taherkhani, Sean Vitousek, Patrick Barnard, Neil Frazer, Tiffany Anderson, and Charles Fletcher. Sea-level rise exponentially increases coastal flood frequency. *Scientific Reports*, 10, 04 2020. 1

[33] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 5

[34] Zhecheng Wang, Rajanie Prabha, Tianyuan Huang, Jiajun Wu, and Ram Rajagopal. Skyscript: A large and semantically diverse vision-language dataset for remote sensing, 2023. 2

[35] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. 5

[36] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models, 2024. 5

[37] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE geoscience and remote sensing magazine*, 5(4):8–36, 2017. 1

A. CoastBench Dataset Description

A.1. Spectral Channels

The dataset is based on Sentinel-2 multi-spectral imagery combined with additional topographical information. The following 12 channels are included in the dataset:

Channel Name	Description
Blue	Sentinel-2 Band 2 (490 nm)
Green	Sentinel-2 Band 3 (560 nm)
Red	Sentinel-2 Band 4 (665 nm)
NIR	Sentinel-2 Band 8 (842 nm)
SWIR16	Sentinel-2 Band 11 (1610 nm)
SWIR22	Sentinel-2 Band 12 (2190 nm)
COP DEM GLO	Copernicus Global Digital Elevation Model (30m)
DeltaDTM	High-resolution Deltares Digital Terrain Model
NDWI	Normalized Difference Water Index
MNDWI	Modified Normalized Difference Water Index
NDVI	Normalized Difference Vegetation Index
NDMI	Normalized Difference Moisture Index

Table 6. Spectral and topographic channels used in the dataset.

A.2. Label Classes

The dataset contains multi-label annotations for coastal and shore types, along with built environment and defence presence. The labels are divided into four groups:

- **Coastal Types (8 Classes)**
  - Cliffed or Steep
  - Moderately Sloped
  - Bedrock Plain
  - Sediment Plain
  - Dune
  - Wetland
  - Inlet
  - Engineered Structures
- **Shore Types (5 Classes)**
  - Sandy Gravel or Small Boulder
  - Muddy Sediments
  - Rocky Shore Platform or Large Boulders
  - Ice or Tundra
  - No Sediment or Shore Platform
- **Built Environment (Binary)**: Presence of human-made structures.
- **Coastal defence (Binary)**: Presence of protective coastal infrastructure.

All labels were manually annotated by Deltares coastal experts based on visual interpretation of multi-year cloud-free composite images.

B. Baseline hyperparameter sweep details

B.1. Sweep configuration

Using Weights and Biases, a hyperparameter sweep using Bayes algorithm was performed to find optimal parameters for the CNN baseline. Parameters that were swept with corresponding values are shown in Table 7.

Hyperparameter	Value	Distribution
Batch Size	$8 - 256$	Fixed values
RandomRotate90 p	$0 - 1$	Uniform
HorizontalFlip p	$0 - 1$	Uniform
VerticalFlip p	$0 - 1$	Uniform
Epochs	$25 - 100$	Uniform
CNN Encoder	vgg16 resnet34 resnet50 efficientnet_b0 efficientnet_b2 convnext_tiny eca_nfnet_l0 mobilenetv3_small_100	Fixed values
Optimizer	Adam AdamW	Fixed values
Learning Rate	$0.000001 - 0.01$	Log uniform
LRScheduler (T_{init})	$25 - 100$	Uniform
LRWarmup (LR_{init})	$0.000000001 - 0.00001$	Log uniform
LRWarmup (T)	$1 - 5$	Uniform

Table 7. Hyperparameters included in the CNN-based baseline sweep.

B.2. Sweep results

After a total of 195 sweep runs, the sweep converged to an average F_1 of 0.720. This optimal run will be named *resnet50* for future identification. The sweep progression is visualized in Figure 14 and a parallel coordinate plot showing relation between parameters and scores is listed in Figure 13. The hyperparameters of the optimal solution with the highest F_1 score can be observed in Table 8.

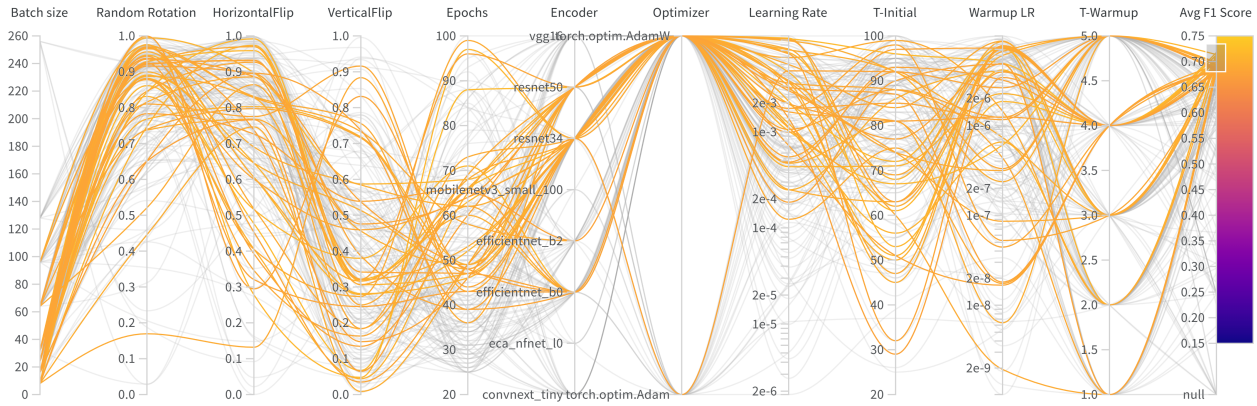


Figure 13. Parallel coordinate plot of the CNN-baseline hyperparameter sweep, showing the relationship between hyperparameter choices and the resulting average F_1 score. The orange lines represent the top-performing configurations, while gray lines indicate lower-performing runs. Clear patterns emerge, with higher F_1 scores associated with lower batch sizes, high RandomRotate probability, AdamW optimizer. and ResNet encoders.

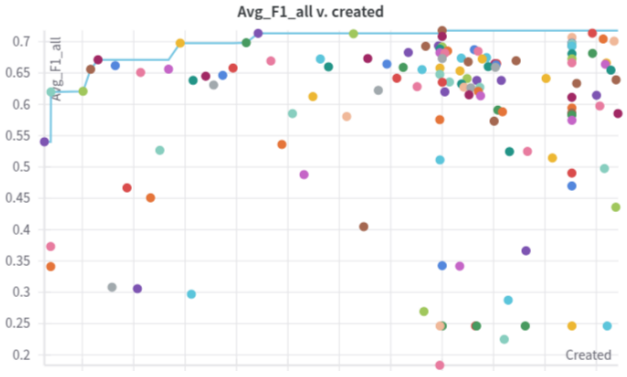


Figure 14. Progression of the average F_1 score during the CNN-baseline sweep, converging to a best score of 0.720 as the hyperparameter search explores the configuration space.

B.3. CNN Model Performance

Figures 15a – 17a show the out-of-fold predictions of the *resnet50* configuration in the form of confusion matrices. The same configuration was also trained using only RGB bands, referred to as *resnet50-rgb*.

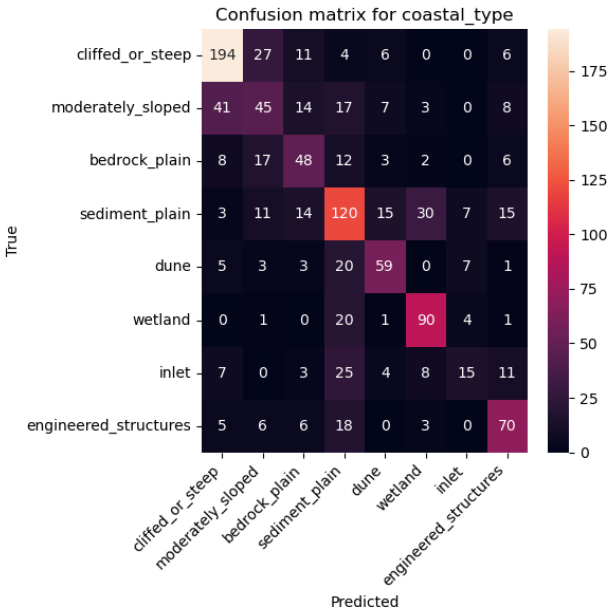
Figure 15 presents the confusion matrices for coastal type classification, comparing the full spectral model (*resnet50*) with the rgb-only model (*resnet50-rgb*).

Figure 16 displays the confusion matrices for shore type classification, showing predictions made by both configurations.

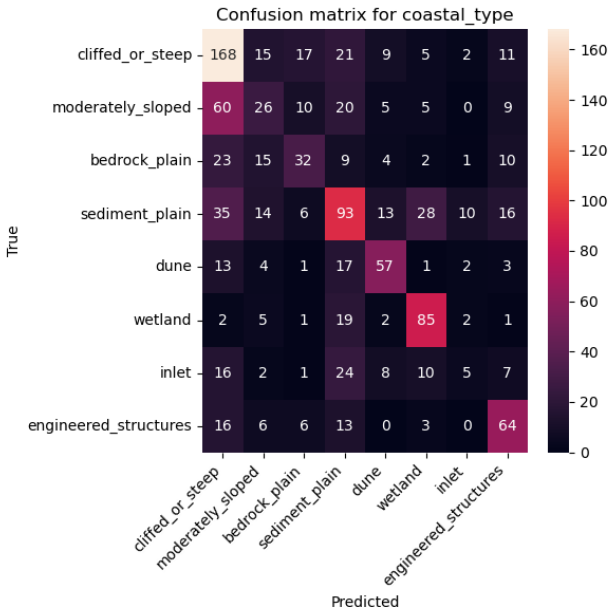
Figure 17 contains the confusion matrices for built environment and coastal defence classification, illustrating how both models classify these categories.

Hyperparameter	
Batch Size	16
RandomRotate90 p	0.881
HorizontalFlip p	0.477
VerticalFlip p	0.067
Epochs	88
CNN Encoder	resnet50
Optimizer	AdamW
Learning Rate	0.009
LRScheduler (T_{init})	96
LRWarmup (LR_{init})	$4.903 \cdot 10^{-6}$
LRWarmup (T)	4

Table 8. Optimal hyperparameters from the CNN-based baseline sweep.

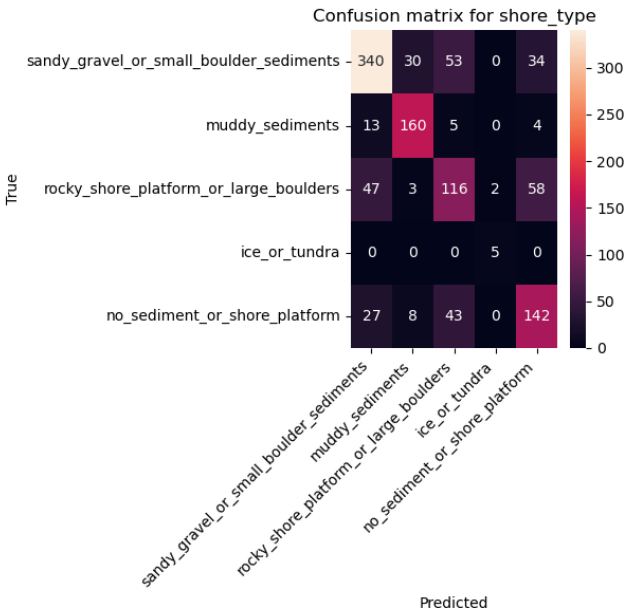


(a) *resnet50* coastal type classification.

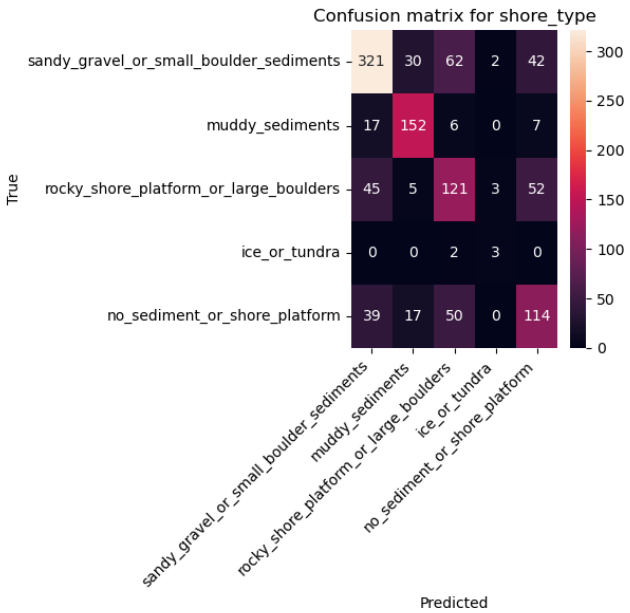


(b) *resnet50-rgb* coastal type classification.

Figure 15. Confusion matrices for coastal type classification.

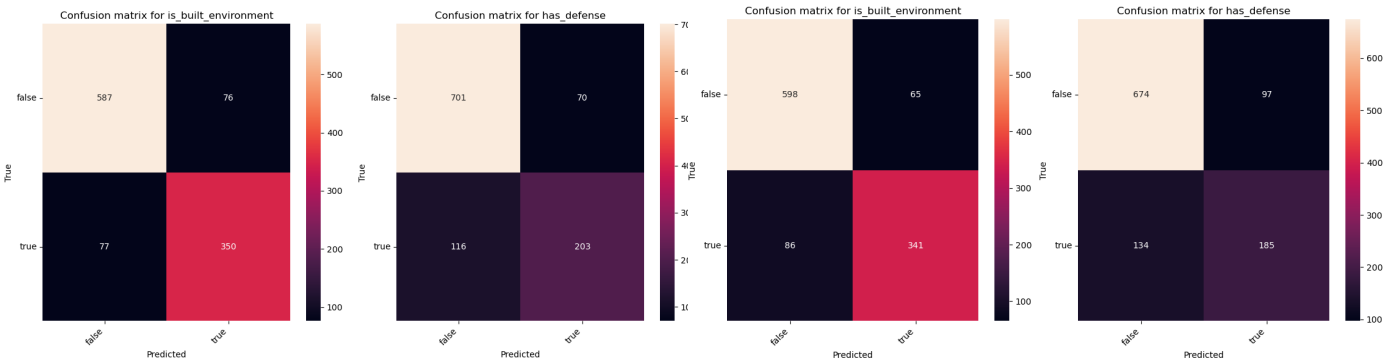


(a) *resnet50* shore type classification.



(b) *resnet50-rgb* shore type classification.

Figure 16. Confusion matrices for shore type classification.



(a) *resnet50* built environment and coastal defence classification.

(b) *resnet50-rgb* built environment and coastal defence classification.

Figure 17. Confusion matrices for built environment and coastal defence classification.

C. MLLM Prompting Methods

This section provides an overview of the prompting strategies used in section 5. A smaller experiment evaluates the impact of different numbers of examples (n) on the classification performance of GPT-4o. As shown in Figure 18, the F_1 score varies with n, with $n = 10$ yielding the highest performance.

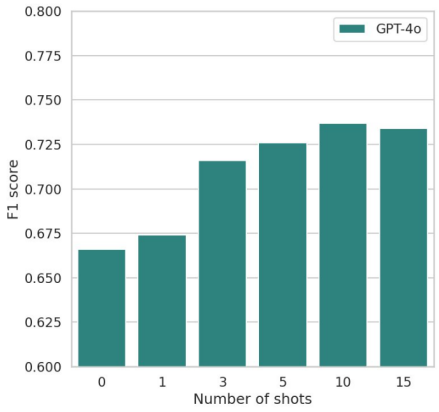


Figure 18. Average F_1 scores of the GPT-4o model on few-shot classification for 50 random samples from the dataset, evaluated with different numbers of shots (n). The best performance is observed at $n = 10$.

C.1. MLLM Prompting Versions

This section provides the detailed prompts used for the models described in Section 5. The different versions correspond to the structured zero-shot approach (V_1), the Chain-of-Thought enhanced version ($V_{1.1}$), and the multi-modal few-shot approach (V_2).

V_1: Zero-shot prompting uses the system prompt C (Listing 1) and user prompt Q (Listing 2) to infer the answer A from the coastal transect image I. $V_{1.1}$: Chain-of-Thought prompting extends V_1 by appending additional reasoning R through the system prompt in Listing 3, which helps the model generate intermediate reasoning chains before producing A. V_2: Few-shot prompting augments the system prompt with n examples. For this, some additional context was appended, shown in Listing 4.

```
You are a vision assistant given a
    classification task specializing in
    coastal remote sensing analysis.
The classification focuses on four key
    attributes:

- Shore Type: Describes the material
    composing the shore (e.g., sandy
    sediments, rocky formations, or
    muddy sediments).
- Coastal Type: Refers to the
    geomorphological features of the
    coast, which may be natural (e.g.,
    cliffs, dunes) or human-influenced (
    e.g., engineered structures).
- Built Environment: Indicates whether
    the coastal area is dominated by
    human-made structures or remains
    largely natural.
- Defenses: Determines whether coastal
    defense structures (e.g., sea walls,
     breakwaters) are present to protect
     against erosion and flooding.

Please answer all the following 4
    questions about the image. Focus on
    the center of your screen.

VERY IMPORTANT: Make sure to ALWAYS
    answer the question in this format:
A1: {answer}
A2: {answer}
A3: {answer}
A4: {answer}.

Do not move away from this format.
```

Listing 1. V_1: System Prompt C

```
Q1: What is the coastal type of this
    image?
Choose one of the following for coastal
    type: ['cliffed_or_steep', '
    moderately_sloped', 'bedrock_plain',
     'sediment_plain', 'dune', 'wetland
    ', 'inlet', 'engineered_structures
    '].

Q2: What is the shore type of this image
    ?
Choose one of the following for shore
    type: ['
    sandy_gravel_or_small_boulder_sediments
    ', 'muddy_sediments', '
    rocky_shore_platform_or_large_boulders
    ', 'ice_or_tundra', '
    no_sediment_or_shore_platform'].

Q3: Is there a built environment in this
     image? Please answer with yes or no
    .

Q4: Is there coastal defense in this
    image? Please answer with yes or no.
```

Listing 2. V_1: User Prompt Q

```
You can give your explanation and
    thoughtful reasoning on what you see
     after answering the question.
fgg
```

Listing 3. $V_{1.1}$: Chain-of-Thought Reasoning R (appended to C)

```
Below you can see 10 examples of images
    with their corresponding labels.
    Please use these examples to guide
    your answers.
```

Listing 4. V_2: Few-shot Context Extension

C.2. Few-shot selection algorithm

Listing 5 shows the pseudocode used for the *greedy-balanced* few-shot prompting algorithm.

```
function greedy_balanced_diverse_samples(n,
    seed=42)
    set random seed to seed
    initialize current_cube from dataset
    extract features for coastal_type,
        shore_type, built_environment,
        defence_presence

    initialize feature_counts for coastal,
        shore, built, defence to 0
    initialize selected_indices as an empty
        list
    initialize remaining_indices as the full
        list of sample indices

    for i = 1 to n do:
        set max_score to -1
        initialize candidates as an empty
            list

        for each idx in remaining_indices do:
            feature = features[idx]
            score = 0

            // Reward diversity: new feature
                values
            if coastal_counts[feature[0]] ==
                0:
                score += 1
            if shore_counts[feature[1]] == 0:
                score += 1
            if built_counts[feature[2]] == 0:
                score += 1
            if defence_counts[feature[3]] ==
                0:
                score += 1

            // Reward balance:
                underrepresented feature
                values
            score += 1 / (coastal_counts[
                feature[0]] + 1)
            score += 1 / (shore_counts[
                feature[1]] + 1)
            score += 1 / (built_counts[
                feature[2]] + 1)
            score += 1 / (defence_counts[
                feature[3]] + 1)

            if score > max_score then:
                max_score = score
                candidates = [idx]
            else if score == max_score then:
                append idx to candidates

        // Randomly choose among candidates
            with the highest score
        chosen = random.choice(candidates)
        append chosen to selected_indices
        remove chosen from remaining_indices

        // Update feature counts
        chosen_feature = features[chosen]
        coastal_counts[chosen_feature[0]] +=
            1
        shore_counts[chosen_feature[1]] += 1
        built_counts[chosen_feature[2]] += 1
        defence_counts[chosen_feature[3]] +=
            1

    return selected_indices
```

Listing 5. Greedy Balanced Selection Algorithm

D. MLLM Prompting Results

Table 9 shows the results of the 3 models evaluated on 3 different prompting techniques in section 5. Figure 19 shows a scatterplot with the relationship between diversity and balance with F_1 scores for different categories (coast, shore, built, defence, and all).

Version	Method	Diversity	Balance	F1 Coast	F1 Shore	F1 Built	F1 Defence	F1 All
GPT-4o								
v1	-	-	-	0.4049	0.5939	0.7280	0.8266	0.6384
v1_1	-	-	-	0.4194	0.5925	0.7125	0.8169	0.6355
v2	Random	0.588	0.332	0.442 ± 0.01	0.619 ± 0.01	0.823 ± 0.02	0.814 ± 0.04	0.673 ± 0.02
v2	Brute	0.882	0.638	0.451 ± 0.00	0.610 ± 0.00	0.855 ± 0.00	0.832 ± 0.01	0.687 ± 0.00
v2	Greedy	1.000	0.669	0.447 ± 0.03	0.597 ± 0.01	0.827 ± 0.00	0.822 ± 0.01	0.673 ± 0.01
v2	Greedy_Balanced	1.000	0.683	0.465 ± 0.00	0.605 ± 0.01	0.823 ± 0.04	0.830 ± 0.00	0.681 ± 0.01
LLaMA-3.2-11B-Vision-Instruct								
v1	-	-	-	0.1920	0.3583	0.6301	0.6508	0.4578
v1_1	-	-	-	0.1535	0.3354	0.6781	0.6600	0.4536
v2	Random	0.588	0.332	0.108 ± 0.02	0.286 ± 0.01	0.496 ± 0.05	0.597 ± 0.02	0.290 ± 0.11
v2	Brute	0.882	0.638	0.105 ± 0.02	0.280 ± 0.01	0.480 ± 0.04	0.593 ± 0.01	0.323 ± 0.04
v2	Greedy	1.000	0.669	0.109 ± 0.01	0.280 ± 0.02	0.483 ± 0.04	0.595 ± 0.01	0.323 ± 0.05
v2	Greedy_Balanced	1.000	0.683	0.112 ± 0.03	0.287 ± 0.02	0.482 ± 0.04	0.589 ± 0.01	0.326 ± 0.06
Qwen-2.5-VL-7B-Instruct								
v1	-	-	-	0.1675	0.4209	0.7515	0.6752	0.4824
v1_1	-	-	-	0.1749	0.4368	0.5328	0.4543	0.3991
v2	Random	0.588	0.332	0.164 ± 0.04	0.486 ± 0.02	0.745 ± 0.04	0.757 ± 0.01	0.538 ± 0.01
v2	Brute	0.882	0.638	0.153 ± 0.03	0.491 ± 0.03	0.755 ± 0.03	0.755 ± 0.01	0.539 ± 0.02
v2	Greedy	1.000	0.669	0.183 ± 0.03	0.496 ± 0.03	0.747 ± 0.05	0.735 ± 0.03	0.540 ± 0.02
v2	Greedy_Balanced	1.000	0.683	0.204 ± 0.03	0.515 ± 0.02	0.730 ± 0.04	0.723 ± 0.03	0.543 ± 0.02

Table 9. F_1 scores with standard deviations of the 3 models evaluated on 3 different prompting techniques.

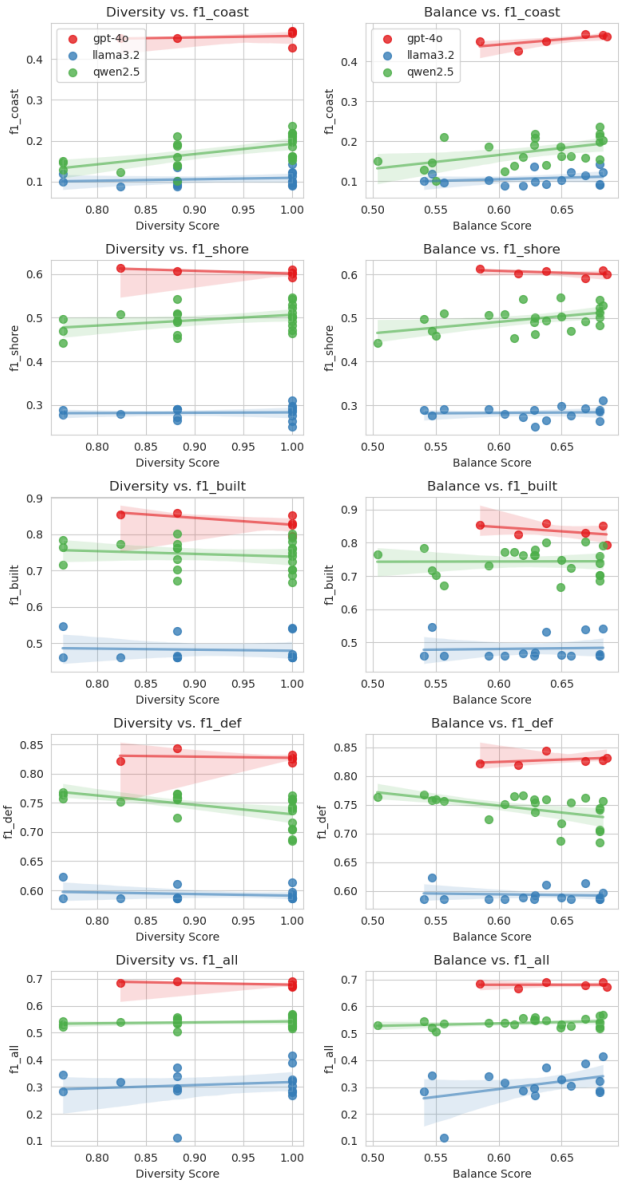


Figure 19. Scatter plots showing the relationship between diversity score (left column) and balance score (right column) with F_1 scores for different categories (coast, shore, built, defence, and all). The three models (GPT-4o, LLaMA 3.2, and Qwen 2.5) are represented by different colors. Each plot includes a regression line representing the mean trend (solid line) for each model, with the shaded region indicating the 95% confidence interval (CI). A positive slope suggests a correlation between the metric and F_1 score, while a flatter or negative slope indicates little to no relationship.

E. MLLM Finetuning

This appendix provides a detailed overview of the fine-tuning process applied to the *qwen2.5* model for coastal classification using multi-modal input data. Fine-tuning was performed to explore the impact of training a vision-language model not just on standard RGB imagery, but also on the full 12-channel satellite input. Table 10 outlines the complete hyperparameter configuration used during this process. A modified training setup, denoted as *qwen2.5-rgb*, was also introduced, where the input channels were reduced to 3 (RGB) to serve as a comparison baseline. This enables a controlled evaluation of the added value of multi-spectral and elevation information when embedded directly into the vision encoder.

To better understand the performance of the fine-tuned MLLMs, we report confusion matrices for each of the key classification targets: coastal type, shore type, and presence of built environment and coastal defence structures. These results are visualised in Figures 20a, 21a and 22 comparing both the 12-channel and RGB-only variants of the model. These visualisations not only highlight the differences in class-wise predictive accuracy but also provide insights into which categories benefit most from enriched input data.

F. Ablation study

In this appendix we present the full tabulated results of our ablation experiments. Table 11 reports the per-feature (individual-channel) ablations, while Table 12 summarises the group-level ablation (dropping entire feature subsets). Each table lists, for every run variant, the F1 scores on the Coast, Shore, Built, and Defence labels. as well as the overall F1 ("F1 All"), the absolute drop in overall F1 ("F1 All Drop"), and the relative percentage drop ("F1 All % Drop").

G. Examples of CoastBench misclassifications

To better understand model limitations and the inherent challenges in coastal classification, we present a few examples of representative failure cases from the CoastBench dataset. These examples were selected not because the models made clear mistakes, but precisely because the ground truth itself becomes debatable upon closer inspection. In several instances, model predictions (and expert disagreements) reveal alternative interpretations, often arising from transitional coastal morphologies or allowing for multiple classes in one transect.

H. Examples of Finetuned Qwen2.5 interactions

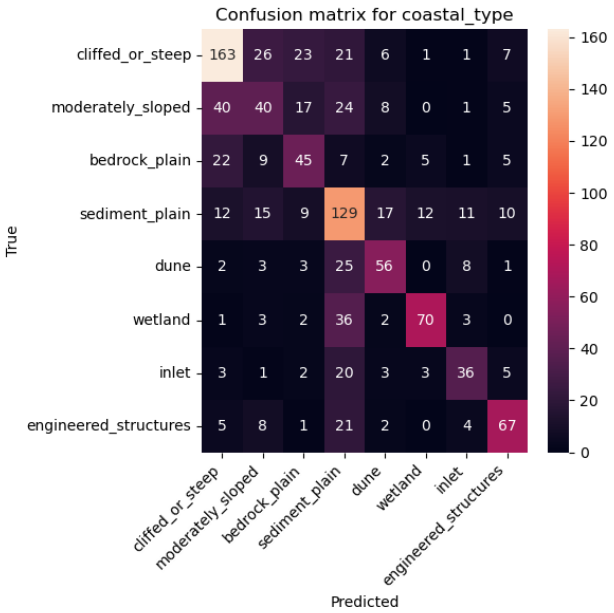
This section contains some examples of interactions with finetuned *qwen2.5-rgb* using Gradio [1].

Table 10. Training parameters for Qwen2.5-VL-7B-Instruct with 12-channel input

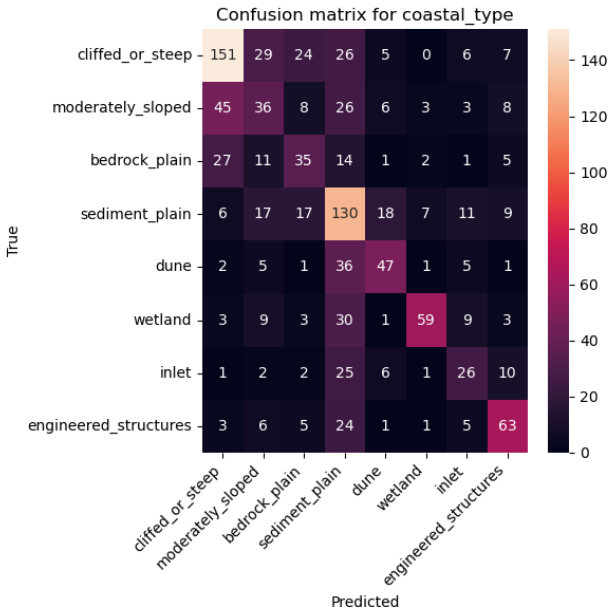
Parameter	Value
model_id	Qwen2.5-VL-7B-Instruct
channels	12
lora_enable	True
vision_lora	True
lora_rank	8
lora_alpha	32
lora_dropout	0.05
lora_namespan_exclude	[lm_head, embed_tokens]
num_lora_modules	-1
freeze_vision_tower	True
freeze_llm	True
tune_merger	True
bf16	True
fp16	False
disable_flash_attn2	True
gradient_checkpointing	False
gradient_accumulation_steps	4
per_device_train_batch_size	4
per_device_eval_batch_size	2
num_train_epochs	60
learning_rate	1e-4
merger_lr	1e-5
vision_lr	1e-6
weight_decay	0.15
warmup_ratio	0.03
lr_scheduler_type	cosine
min_pixels / max_pixels	$200 \times 28 \times 28$
eval_strategy	steps
eval_steps	250
eval_accumulation_steps	1
predict_with_generate	True
save_strategy	no
save_steps	500
save_total_limit	10
logging_strategy	steps
logging_steps	1
report_to	wandb
lazy_preprocess	True
dataloader_num_workers	1
dataloader_drop_last	True
tf32	True
augment	True

H.1. Out-Of-Data distribution examples

Here we can see some manual experimentation with coastal areas outside of CoastBench to get an impression of the generalisation to other images. Although it cannot be

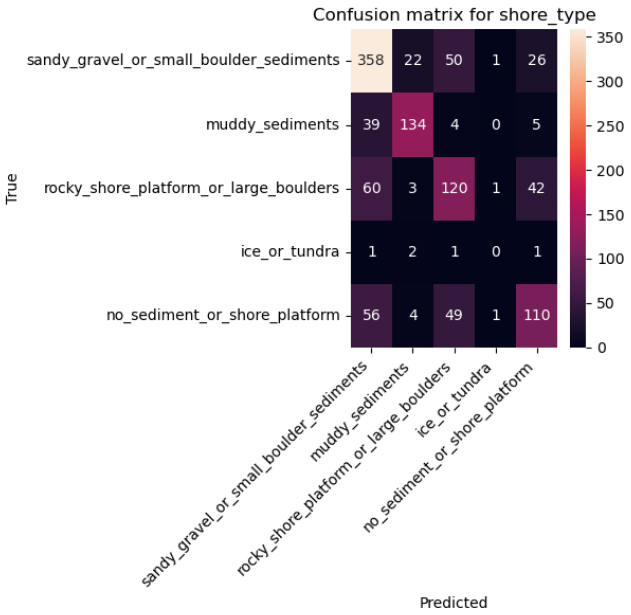


(a) *qwen2.5* coastal type classification.

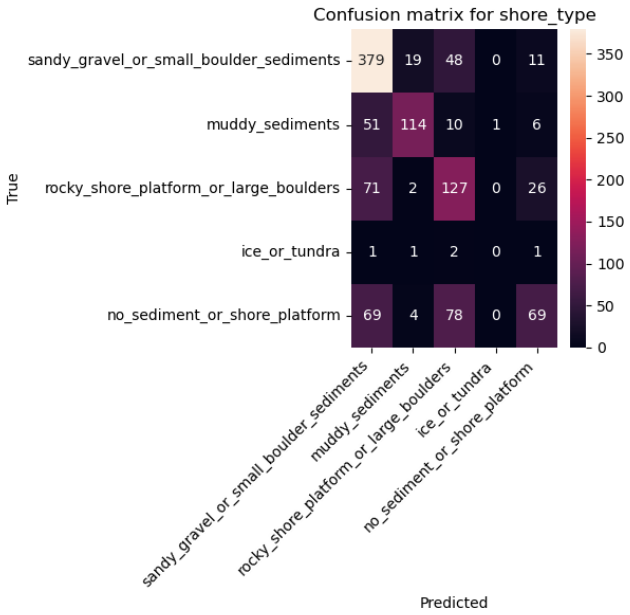


(b) *qwen2.5-rgb* coastal type classification.

Figure 20. Confusion matrices for coastal type classification.



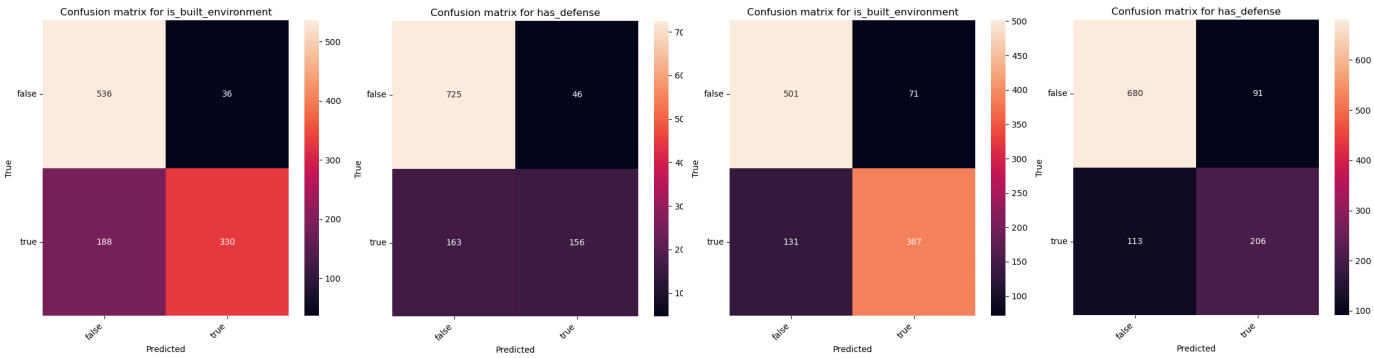
(a) *qwen2.5* shore type classification.



(b) *qwen2.5-rgb* shore type classification.

Figure 21. Confusion matrices for shore type classification.

ruled out that the examples were seen during the model's pretraining, we can still observe the model's structural consistency under dataset shift.



(a) *qwen2.5* built environment and coastal defence classification.

(b) *qwen2.5-rgb* built environment and coastal defence classification.

Figure 22. Confusion matrices for built environment and coastal defence classification.

Run Name	F1 Coast	F1 Shore	F1 Built	F1 Defence	F1 All	F1 All Drop	F1 All % Drop
cnn-best	0.577 ± 0.003	0.694 ± 0.007	0.859 ± 0.000	0.824 ± 0.002	0.719 ± 0.002	0.000	0.000
cnn-best-no-swir22	0.569 ± 0.001	0.696 ± 0.010	0.847 ± 0.021	0.828 ± 0.007	0.711 ± 0.016	0.008 ± 0.030	1.095
cnn-best-no-swir16	0.546 ± 0.002	0.684 ± 0.019	0.855 ± 0.007	0.818 ± 0.003	0.704 ± 0.006	0.015 ± 0.025	2.114
cnn-best-no-nir	0.557 ± 0.003	0.687 ± 0.013	0.851 ± 0.005	0.824 ± 0.008	0.707 ± 0.008	0.012 ± 0.031	1.713
cnn-best-no-red	0.551 ± 0.006	0.680 ± 0.005	0.858 ± 0.003	0.820 ± 0.004	0.710 ± 0.008	0.010 ± 0.020	1.329
cnn-best-no-green	0.567 ± 0.014	0.670 ± 0.009	0.852 ± 0.015	0.828 ± 0.006	0.710 ± 0.009	0.009 ± 0.019	1.300
cnn-best-no-blue	0.552 ± 0.004	0.664 ± 0.006	0.847 ± 0.007	0.819 ± 0.003	0.699 ± 0.001	0.020 ± 0.014	2.737
cnn-best-no-NDWI	0.555 ± 0.005	0.682 ± 0.008	0.847 ± 0.004	**0.809** ± 0.003	0.700 ± 0.006	0.020 ± 0.023	2.719
cnn-best-no-NDVI	0.566 ± 0.011	0.676 ± 0.004	0.853 ± 0.004	0.830 ± 0.010	0.711 ± 0.003	0.008 ± 0.022	1.160
cnn-best-no-NDMI	0.575 ± 0.004	0.652 ± 0.008	0.860 ± 0.009	**0.809** ± 0.006	0.701 ± 0.008	0.018 ± 0.026	2.525
cnn-best-no-MNDWI	0.570 ± 0.012	0.672 ± 0.006	0.862 ± 0.005	0.824 ± 0.008	0.710 ± 0.007	0.009 ± 0.021	1.292
cnn-best-no-deltadtm	0.564 ± 0.006	0.685 ± 0.021	0.857 ± 0.003	0.817 ± 0.012	0.707 ± 0.007	0.012 ± 0.025	1.668
cnn-best-no-cop	0.572 ± 0.011	0.670 ± 0.017	0.853 ± 0.003	0.820 ± 0.014	0.708 ± 0.016	0.012 ± 0.031	1.613

Table 11. Individual-feature ablation results, grouped by feature category (SWIR, RGB, spectral indices, and elevation). Each row shows the mean F_1 score ± standard deviation across two seeds and five folds (10 samples total). We report class-wise scores (Coast, Shore, Built, Defence) as well as overall F1 All. The columns F1 All Drop and F1 All % Drop indicate the absolute and relative decline in overall F_1 compared to the full model (cnn-best).

Run Name	F1 Coast	F1 Shore	F1 Built	F1 Defence	F1 All	F1 All Drop	F1 All % Drop
cnn-best	0.577 ± 0.003	0.694 ± 0.007	0.859 ± 0.000	0.824 ± 0.002	0.719 ± 0.002	0.000	0.000
cnn-best-no-infrared	0.559 ± 0.003	0.664 ± 0.005	0.854 ± 0.017	0.821 ± 0.004	0.700 ± 0.006	0.020 ± 0.017	2.728
cnn-best-no-rgb	0.564 ± 0.021	0.662 ± 0.033	**0.837** ± 0.005	0.825 ± 0.002	0.704 ± 0.020	0.016 ± 0.023	2.155
cnn-best-no-spectral	0.556 ± 0.005	0.666 ± 0.032	0.863 ± 0.003	**0.807** ± 0.011	0.705 ± 0.013	0.014 ± 0.028	2.016
cnn-best-no-elevation	**0.502** ± 0.005	0.698 ± 0.007	0.863 ± 0.011	0.828 ± 0.011	0.705 ± 0.011	0.014 ± 0.016	1.984

Table 12. Group-level ablation results showing the effect of removing entire feature categories (Infrared, RGB, Spectral Indices, and Elevation Maps) from the input data. Each row reports the mean F_1 score ± standard deviation, averaged over two seeds and five folds (10 evaluations total). We present class-wise F_1 scores as well as the overall F_1 (All), along with the absolute and relative performance drops compared to the full model (cnn-best).

True Labels
- **Shore Type:** muddy_sediments
- **Coastal Type:** wetland
- **Is Built Environment:** false
- **Has Defense:** false

Antonio | Juul | cnn-best (0.72) Predictions
- **Shore Type:** muddy_sediments | muddy_sediments | muddy_sediments
- **Coastal Type:** inlet | inlet | wetland
- **Is Built Environment:** false | false | false
- **Has Defense:** false | false | false

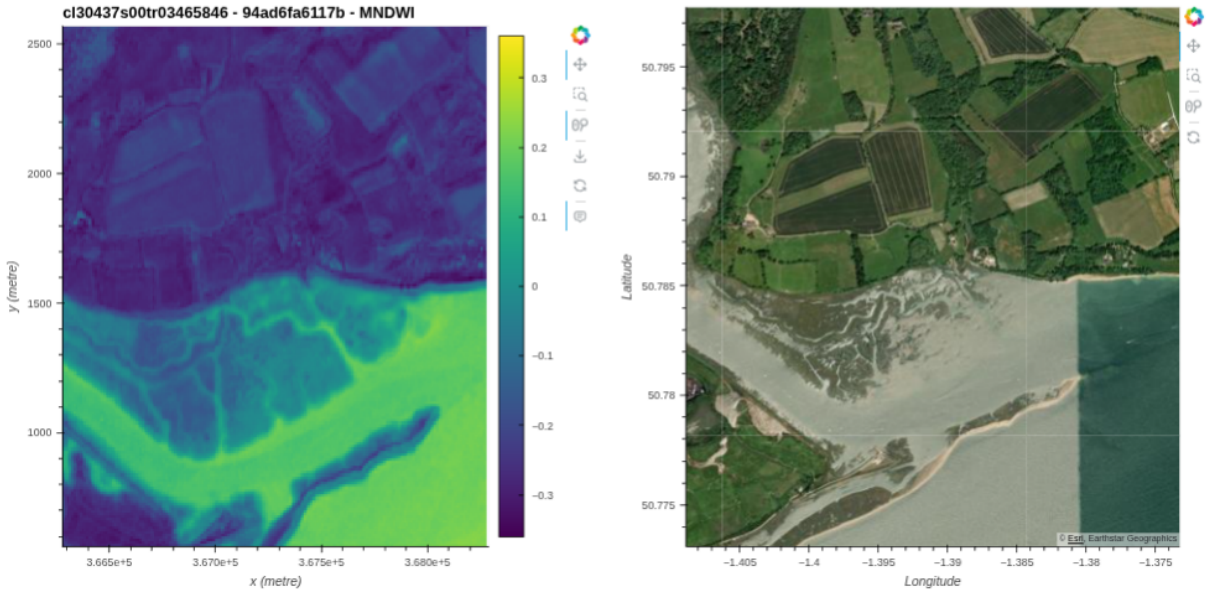


Figure 23. Example of a visually ambiguous transect where model and expert predictions diverge. While the ground truth labels this area as a wetland, both experts predicted an inlet. The satellite image reveals a tidal river mouth with surrounding vegetated zones, making it a borderline case between an inlet and a wetland. This underscores the inherent subjectivity in coastal type labelling, even when full spatial context is available.

True Labels
- **Shore Type:** rocky_shore_platform_or_large_boulders
- **Coastal Type:** moderately_sloped
- **Is Built Environment:** false
- **Has Defense:** true

Antonio | Juul | cnn-best (0.72) Predictions
- **Shore Type:** rocky_shore_platform_or_large_boulders | rocky_shore no_sediment_or_shore_platform
- **Coastal Type:** cliffed_or_steep | cliffed_or_steep | engineered
- **Is Built Environment:** true | true | false
- **Has Defense:** true | true | true

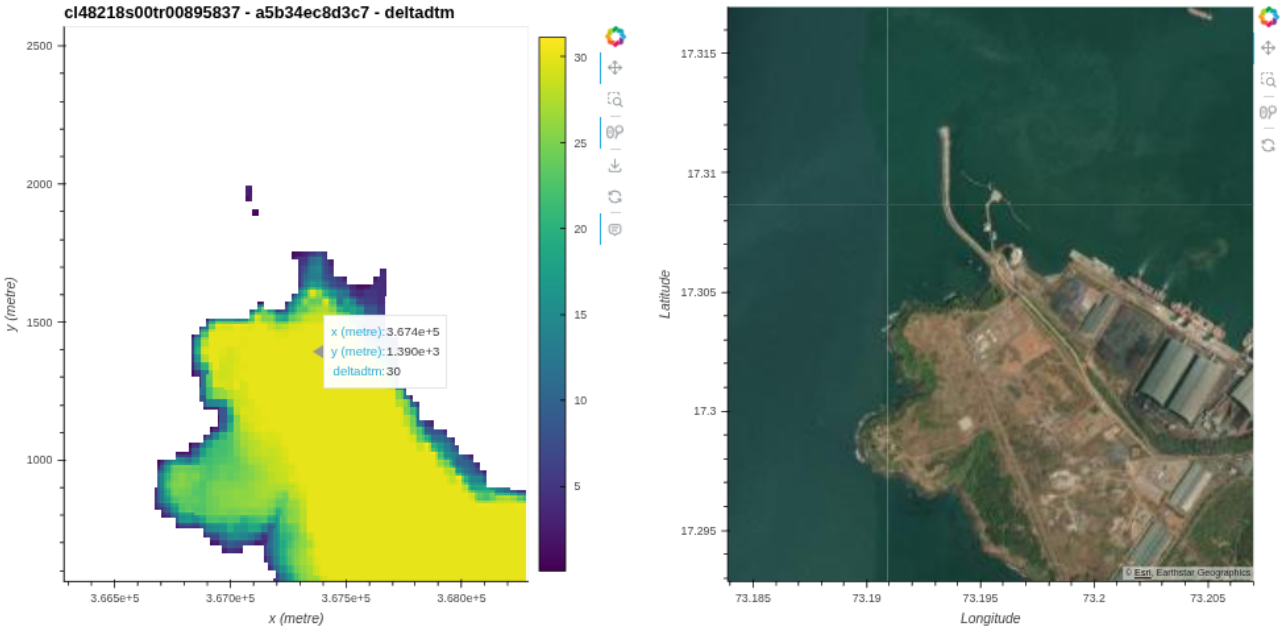


Figure 24. Example of a mislabeled coastal transect that challenges both experts and models. The DeltaDTM elevation reaches $\tilde{3}0$, indicating a steep, cliffed coast, a conclusion independently supported by both expert annotators and the CNN model, despite the ground truth label being moderately_sloped. In the satellite image, we clearly observe engineered structures, including a pier, docked ships, industrial hangars, and a coastal access road, strongly suggesting the presence of a built environment. However, the reference label annotates this as not built.

True Labels
- **Shore Type:** rocky_shore_platform_or_large_boulders
- **Coastal Type:** dune
- **Is Built Environment:** false
- **Has Defense:** false

Antonio | Juul | cnn-best (0.72) Predictions
- **Shore Type:** sandy_gravel_or_small_boulder_sediments | sandy_gravel_or_small_boulder_sediments
- **Coastal Type:** sediment_plain | sediment_plain | dune
- **Is Built Environment:** false | false | false
- **Has Defense:** false | false | false

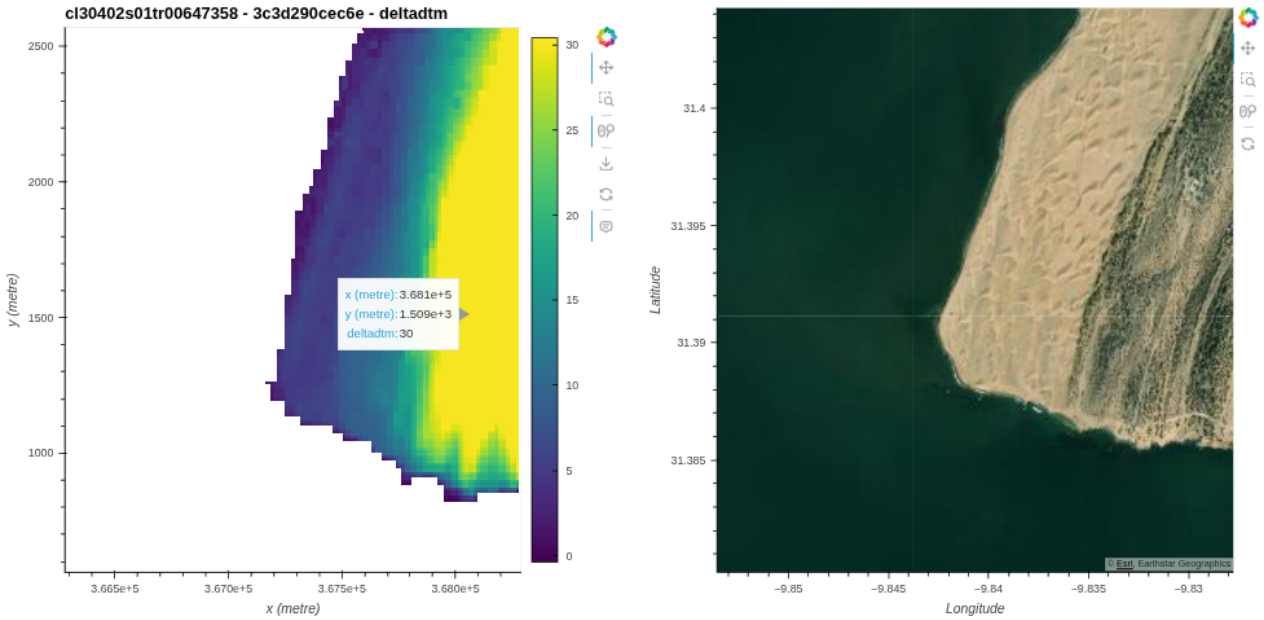


Figure 25. Example of a visually clear sandy coastline that contradicts the ground truth label. The satellite image and DeltaDTM elevation (reaching 30) indicate a classic dune landscape, with sandy beach ridges and vegetated zones on the inland side. Both experts and the CNN model correctly predict sandy gravel shore types However, the reference label assigns this area a rocky shore platform, which is inconsistent with both the visual appearance and contextual cues.

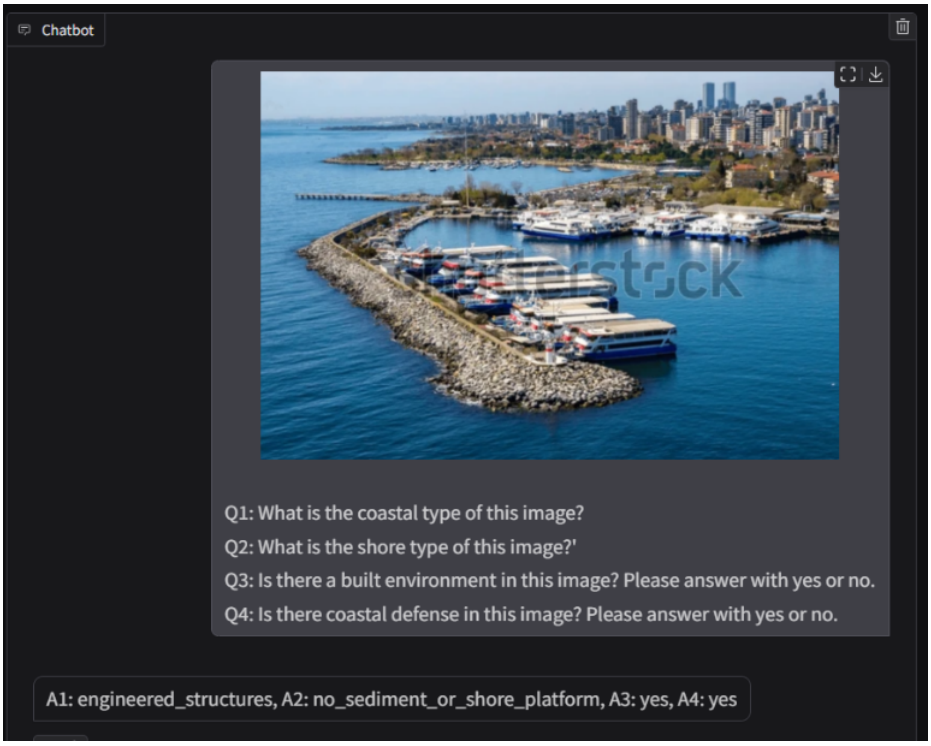


Figure 26. Example of an out-of-distribution (OOD) harbour image not present in the CoastBench dataset. The model *qwen2.5-rgb* was prompted with the standard four coastal classification questions. Despite the absence of domain-specific training on harbours, the model produces structured outputs: A1: engineered_structures, A2: no_sediment_or_shore_platform, A3: yes, A4: yes. These responses are plausible, though no ground truth labels exist for this image. This example highlights the model's ability to produce consistent and contextually reasonable outputs under distributional shift.

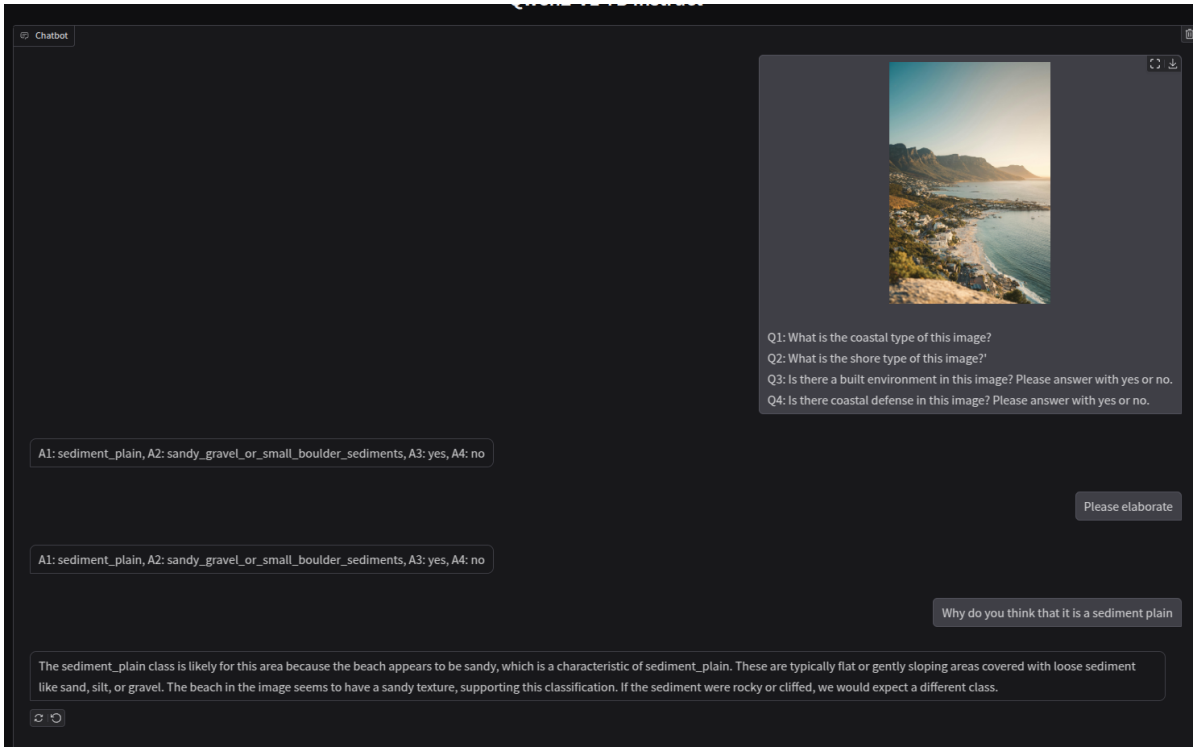


Figure 27. Out-of-distribution (OOD) example depicting a coastline manually sourced from the web and not part of the CoastBench dataset. The image was evaluated using the standard structured zero-shot prompt with the *qwen2.5-RGB* model. The model returned the following predictions: A1: sediment_plain, A2: sandy_gravel_or_small_boulder_sediments, A3: yes, A4: no. When prompted with "Please elaborate," the model initially repeated the exact same structured output without explanation, suggesting that it did not parse the instruction as a follow-up. This behavior highlights a known limitation in multi-turn vision-language interaction: models may fail to contextually link follow-up prompts unless phrased with precise intent. While the predictions here appear plausible, no ground truth is available for verification, and such examples serve only as qualitative insight into model behavior under dataset shift.

References

[1] Edward B. Barbier. “Valuing Ecosystem Services for Coastal Wetland Protection and Restoration: Progress and Challenges”. In: *Resources* 2.3 (2013), pp. 213–230. ISSN: 2079-9276. DOI: 10.3390/resources2030213. URL: https://www.mdpi.com/2079-9276/2/3/213.

[2] Andrew Short. “Coastal processes and beaches”. In: *Nat. Educ. Knowl.* 3 (Jan. 2012).

[3] Carlos Méndez et al. *Climate Change 2023: Synthesis Report (Full Volume) Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. July 2023. DOI: 10.59327/IPCC/AR6-9789291691647.

[4] Eva M. Lansu et al. “A global analysis of how human infrastructure squeezes sandy coasts”. English. In: *Nature Communications* 15.1 (2024). ISSN: 2041-1723. DOI: 10.1038/s41467-023-44659-0.

[5] Tianze Pang et al. “Coastal erosion and climate change: A review on coastal-change process and modeling”. In: *Ambio* 52 (July 2023). DOI: 10.1007/s13280-023-01901-9.

[6] Marjolijn Haasnoot et al. “Long-term sea-level rise necessitates a commitment to adaptation: A first order assessment”. In: *Climate Risk Management* 34 (2021), p. 100355. ISSN: 2212-0963. DOI: https://doi.org/10.1016/j.crm.2021.100355. URL: https://www.sciencedirect.com/science/article/pii/S221209632100084X.

[7] John A. Church and Neil J. White. “A 20th century acceleration in global sea-level rise”. In: *Geophysical Research Letters* 33.1 (2006). DOI: https://doi.org/10.1029/2005GL024826. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2005GL024826. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2005GL024826.

[8] Xiao Xiang Zhu et al. “Deep learning in remote sensing: A comprehensive review and list of resources”. In: *IEEE geoscience and remote sensing magazine* 5.4 (2017), pp. 8–36.

[9] Floris Reinier Calkoen et al. “Enabling coastal analytics at planetary scale”. In: *Environmental Modelling & Software* 183 (2025), p. 106257. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2024.106257. URL: https://www.sciencedirect.com/science/article/pii/S1364815224003189.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[11] Christopher Brown et al. “Dynamic World, Near real-time global 10 m land use land cover mapping”. In: *Scientific Data* 9 (June 2022), p. 251. DOI: 10.1038/s41597-022-01307-4.

[12] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[13] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[14] J. A. G. Cooper and S. McLaughlin. “Contemporary Multidisciplinary Approaches to Coastal Classification and Environmental Risk Analysis”. In: *Journal of Coastal Research* 14.2 (1998), pp. 512–524. ISSN: 07490208, 15515036. URL: http://www.jstor.org/stable/4298806 (visited on 02/24/2025).

[15] K. Ahrendt et al. “A New Multifunctional Coastal Classification for Ecosystem-Service Assessments”. In: *Natural Resources Conservation and Research* 1.1 (2018), pp. 1–10. DOI: 10.1016/j.envsci.2018.04.005.

[16] Charles Finkl. “Coastal classification: Systematic approaches to consider in the development of a comprehensive scheme”. In: *Journal of Coastal Research* 20 (Dec. 2004), pp. 166–213.

[17] Floris Calkoen et al. “MAPPING COASTAL TYPOLOGY USING PUBLICLY AVAILABLE EARTH OBSERVATION DATA AND DEEP NEURAL NETWORKS”. In: *Coastal Engineering Proceedings* (Sept. 2023), p. 158. DOI: 10.9753/icce.v37.management.158.

[18] Eirini Skrimizea, Dimitris Papakonstantinou, and Angelos Siolas. “Integrated Coastal Zone Management Method for Part of the South-Western Attica’s Coastal Area”. In: Jan. 2015.

[19] Christopher Sharples. “The Smartline - an effective coastal data mapping format”. In: (Jan. 2008). URL: https://figshare.utas.edu.au/articles/conference_contribution/The_Smartline_-_an_effective_coastal_data_mapping_format/23132483.

[20] European Commission. *Living with Coastal Erosion in Europe: Sediment and Space for Sustainability*. Tech. rep. Brussels, Belgium: Directorate-General for Environment, 2004. URL: https://www.eurosion.org/reports-online/.

[21] Romy Hulskamp et al. “Global distribution and dynamics of muddy coasts”. In: *Nature communications* 14 (Dec. 2023), p. 8259. DOI: 10.1038/s41467-023-43819-6.

[22] Kinh Bac Dang et al. “A Convolutional Neural Network for Coastal Classification Based on ALOS and NOAA Satellite Data”. In: *IEEE Access* 8 (2020), pp. 11824–11839. DOI: 10.1109/ACCESS.2020.2965231.

[23] European Space Agency. *Sentinel Overview*. https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-2. Accessed 2025-05-06. 2023.

[24] European Space Agency. *Copernicus Global Digital Elevation Model*. Accessed: 2025-03-20. 2024. URL: https://doi.org/10.5069/G9028PQB.

[25] Maarten Pronk et al. “DeltaDTM: A global coastal digital terrain model”. English. In: *Scientific Data* 11 (2024). ISSN: 2052-4463. DOI: 10.1038/s41597-024-03091-9.

[26] Rema Abdusamea. “The Importance of the Normalized Difference Vegetation Index (NDVI) and the Use of the ArcGIS to create NDVI Maps”. In: □□□□ □□□□□□ □□□□□□□□□ □□□□□□□□□ (Jan. 2018), p. 1. DOI: 10.37376/1571-000-057-002.

[27] S. K. McFEETERS and. “The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features”. In: *International Journal of Remote Sensing* 17.7 (1996), pp. 1425–1432. DOI: 10.1080/01431169608948714. eprint: https://doi.org/10.1080/01431169608948714. URL: https://doi.org/10.1080/01431169608948714.

[28] Hanqiu Xu and. “Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery”. In: *International Journal of Remote Sensing* 27.14 (2006), pp. 3025–3033. DOI: 10.1080/01431160600589179. eprint: https://doi.org/10.1080/01431160600589179. URL: https://doi.org/10.1080/01431160600589179.

[29] Berca Mihai and Roxana Horoias. “NDMI USE IN RECOGNITION OF WATER STRESS ISSUES, RELATED TO WINTER WHEAT YIELDS IN SOUTHERN ROMANIA”. In: June 2022.

[30] Abien Fred Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. 2019. arXiv: 1803.08375 [cs.NE]. URL: https://arxiv.org/abs/1803.08375.

[31] Sridhar Narayan. “The generalized sigmoid activation function: Competitive supervised learning”. In: *Information Sciences* 99.1 (1997), pp. 69–82. ISSN: 0020-0255. DOI: https://doi.org/10.1016/S0020-0255(96)00200-9. URL: https://www.sciencedirect.com/science/article/pii/S0020025596002009.

[32] Yupeng Chang et al. “A survey on evaluation of large language models”. In: *ACM transactions on intelligent systems and technology* 15.3 (2024), pp. 1–45.

[33] Jawad Nagi et al. “Max-pooling convolutional neural networks for vision-based hand gesture recognition”. In: *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. 2011, pp. 342–347. DOI: 10.1109/ICSIPA.2011.6144164.

[34] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[35] Ashish Vaswani et al. “Attention Is All You Need”. In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[36] Hugo Touvron et al. “LLaMA: Open and Efficient Foundation Language Models”. In: *arXiv preprint arXiv:2302.13971* (2023).

[37] Kartik Kuckreja et al. *GeoChat: Grounded Large Vision-Language Model for Remote Sensing*. 2023. arXiv: 2311.15826 [cs.CV]. URL: https://arxiv.org/abs/2311.15826.

[38] Zhecheng Wang et al. *SkyScript: A Large and Semantically Diverse Vision-Language Dataset for Remote Sensing*. 2023. arXiv: 2312.12856 [cs.CV]. URL: https://arxiv.org/abs/2312.12856.

[39] Shuai Bai et al. *Qwen2.5-VL Technical Report*. 2025. arXiv: 2502.13923 [cs.CV]. URL: https://arxiv.org/abs/2502.13923.

[40] Noam Shazeer. "GLU Variants Improve Transformer". In: *CoRR* abs/2002.05202 (2020). arXiv: 2002.05202. URL: https://arxiv.org/abs/2002.05202.

[41] Biao Zhang and Rico Sennrich. "Root Mean Square Layer Normalization". In: *CoRR* abs/1910.07467 (2019). arXiv: 1910.07467. URL: http://arxiv.org/abs/1910.07467.

[42] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[43] Florian. *The number of parameters of GPT-4O and claude 3.5 sonnet*. Jan. 2025. URL: https://aiexpjourney.substack.com/p/the-number-of-parameters-of-gpt-4o.

[44] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[45] Pengfei Liu et al. "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Computing Surveys* 55.9 (2023), pp. 1–35. URL: https://arxiv.org/abs/2107.13586.

[46] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[47] Jason Wei et al. "Chain of Thought Prompting Elicits Reasoning in Large Language Models". In: *CoRR* abs/2201.11903 (2022). arXiv: 2201.11903. URL: https://arxiv.org/abs/2201.11903.

[48] Takeshi Kojima et al. *Large Language Models are Zero-Shot Reasoners*. 2023. arXiv: 2205.11916 [cs.CL]. URL: https://arxiv.org/abs/2205.11916.