Journal of *Marine Science and Engineering*

MDPI

Article

Evaluation of Deep Learning Models for Polymetallic Nodule Detection and Segmentation in Seafloor Imagery

Gabriel Loureiro [1,*], André Dias [1,2], José Almeida [1,2], Alfredo Martins [1,2] and Eduardo Silva [1,2]

[1] INESCTEC—Institute for Systems and Computer Engineering, Technology and Science, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal; andre.dias@inesctec.pt (A.D.); jose.m.almeida@inesctec.pt (J.A.); alfredo.martins@inesctec.pt (A.M.); eduardo.silva@inesctec.pt (E.S.)

[2] ISEP—School of Engineering, Polytechnic Institute of Porto, Rua Dr. António Bernardino de Almeida 431, 4200-072 Porto, Portugal

* Correspondence: gabriel.s.loureiro@inesctec.pt

Abstract: Climate change has led to the need to transition to clean technologies, which depend on an number of critical metals. These metals, such as nickel, lithium, and manganese, are essential for developing batteries. However, the scarcity of these elements and the risks of disruptions to their supply chain have increased interest in exploiting resources on the deep seabed, particularly polymetallic nodules. As the identification of these nodules must be efficient to minimize disturbance to the marine ecosystem, deep learning techniques have emerged as a potential solution. Traditional deep learning methods are based on the use of convolutional layers to extract features, while recent architectures, such as transformer-based architectures, use self-attention mechanisms to obtain global context. This paper evaluates the performance of representative models from both categories across three tasks: detection, object segmentation, and semantic segmentation. The initial results suggest that transformer-based methods perform better in most evaluation metrics, but at the cost of higher computational resources. Furthermore, recent versions of You Only Look Once (YOLO) have obtained competitive results in terms of mean average precision.

Keywords: deep sea; polymetallic nodules; deep learning; object detection; object segmentation

Academic Editors: Fausto Pedro García Márquez and Marco Cococcioni

Received: 15 January 2025
Revised: 9 February 2025
Accepted: 10 February 2025
Published: 13 February 2025

Citation: Loureiro, G.; Dias, A.; Almeida, J.; Martins, A.; Silva, E. Evaluation of Deep Learning Models for Polymetallic Nodule Detection and Segmentation in Seafloor Imagery. *J. Mar. Sci. Eng.* **2025**, *13*, 344. https://doi.org/10.3390/jmse13020344

1. Introduction

One significant challenge modern society faces is climate change and its effects on the Earth's ecosystems [1,2]. International accords such as the Paris Agreement [3] and the United Nations Secretariat Climate Action Plan 2020–2030 [4] are examples of initiatives that aim to find solutions to reducing our carbon footprint and advancing sustainable practices. In order to achieve these goals, it is necessary to transition to renewable energy and technologies, which, in turn, depend on critical metals. Critical metals are non-combustible materials that fulfil essential functions for energy technologies and have major disruption risks in their supply chain. Examples of these metals include nickel, cobalt, copper, lithium and others that, due to their characteristics, have an enormous capacity for storing, conducting or transmitting energy [5]. However, as surface sources become more strained while demand increases, there has been a growing interest in locating untapped reserves of these resources, particularly those in the deep sea, including polymetallic nodules, cobalt-rich ferromanganese crusts, and polymetallic sulphides [6].

Polymetallic nodules are small rounded or oval mineral deposits scattered across the deep seabed, known for their high concentrations of essential metals such as manganese,

iron, cobalt, nickel, copper, and other rare-earth elements of economic interest [7]. Traditional mining machines typically rely on a collector that sweeps these nodules off the seafloor and pump systems that transport them up to a surface vessel. Hazard assessment studies have already highlighted the immense impact of these technologies on the seabed, including habitat alterations and sediment plumes [8,9]. Besides this, mining efforts must comply with the regulatory standards set by The International Seabed Authority (ISA), guaranteeing the protection of biodiversity. Therefore, the feasibility of resource exploitation depends directly on our ability to identify and map these resources efficiently. Given the extreme environmental conditions in which they are located, such as high pressure, low visibility, uneven terrain, and with marine life present, this task becomes even more challenging. Although box-corers and photographic profiling remain fundamental for large-scale resource assessments, Autonomous Underwater Vehicles (AUVs) or Remotely Operated Vehicles (ROVs) embedded with a multimodal set of sensors allows for resource exploitation by enabling detailed environmental monitoring and informed decision-making. However, the data processing involved is complex and time-consuming due to the large amount of data those sensors provide.

Recent advances in artificial intelligence, particularly deep learning (DL) techniques, provide promising tools for handling extensive data [10]. These methods may enable real-time monitoring and supervision of the seafloor during mining campaigns, thus enabling impact assessment studies to be carried out. For instance, the Trident project aims to conduct a baseline assessment of the deep-sea environment to establish a detailed understanding of seabed conditions before any activity occurs [11]. Using deep learning techniques to process sensor data, such as seafloor imagery, enables rapid the identification and mapping of the polymetallic nodules.

Classic DL architectures use convolutional neural networks (CNNs), which use convolutional filters to detect feature patterns in an image, generating a hierarchical representation of features through successive layers. On the other hand, recent advances have introduced transformer-based models, in which self-attention mechanisms are used to gather global context from an image [12].

This paper aims to evaluate the performance of several CNN and transformer-based models in detecting and segmenting polymetallic nodules in optical images. For this reason, three types of tasks are considered: object detection, instance segmentation, and semantic segmentation. Object detection identifies the locations of nodules, instance segmentation distinguishes individual nodules, and semantic segmentation provides detailed pixel-level seafloor segmentation. Existing studies often address only a single task and employ traditional deep-learning architectures [13–15], without evaluating the recent advances in attention-based or hybrid models. In addition, focusing on a single task may cause researchers to miss important contextual cues. This study expands on prior work by comprehensively analysing several neural network models (convolution-based and transformer-based) over the three tasks, leveraging state-of-the-art architectures based on different approaches and addressing the limitations of these models in terms of their inference performance and computational resource requirements.

This paper is structured as follows: Section 2 provides an overview of related work, focusing on polymetallic nodule detection and segmentation based on deep learning techniques. Next, Section 3 presents the dataset used in this study and describes the annotation and preprocessing techniques applied for the models' training. Section 4 describes the model selection process carried out for each task. This section details the experimental setup, the models' hyperparameters, and the evaluation metrics used. Section 5 then presents a comprehensive evaluation of model performance with quantitative metrics and qualitative examples. Finally, Section 6 presents a discussion of the obtained results and a

comparison of the different methods, highlighting their respective advantages and disadvantages, followed by our conclusions and recommendations for future research directions.

2. Related Work

Our prior work [16] surveyed a wide range of seafloor mapping techniques, including applying deep learning models for the mapping of polymetallic nodules. We have identified that seabed mapping and characterisation have usually relied on acoustic sensors and optical imagery. This paper expands on that work, focusing on approaches related to polymetallic nodules and analysing their performance in terms of detection and segmentation.

In terms of object detection tasks, state-of-the-art architectures are usually presented in the literature. Quintana et al. [17] applied You Only Look Once (YOLO) and Faster R-CNN models for nodule detection. Faster R-CNN outperformed YOLO in all evaluation metrics. However, a direct comparison of the inference times of the two models was not carried out. Their models were evaluated in a pool and in situ [18,19].

Due to their real-time performance and lightweight architecture, several approaches have employed YOLO models extensively. Sun et al. [20] utilised the YOLOv5s model to detect small-sized polymetallic nodules in hyperspectral data. The model served as a reference network to be fused with the Normalized Wasserstein Distance (NWD) and Intersection Over Union (IoU), improving its performance. Conversely, most of the small nodules, with a diameter of less than 5 cm, covered or buried under seafloor sediment did not appear in the image. Similarly, Cui et al. [21] addressed the challenge of picking nodules with a robotic manipulator using an improved YOLOv5 network, integrating a dual loss function based on NWD and IoU. Eight models were trained for comparison, and the average precision, precision, and recall metrics were used. The improved model increased the average precision by 2.3% compared with the classical YOLOv5. However, the more recent model, YOLOv8, presented a better performance.

For segmentation tasks, a Mask R-CNN model pre-trained on the Common Objects in Context (COCO) dataset was used by Dong et al. [22] to segment nodules in optical images. The model performance was also compared to U-Net and Generative Adversarial Network (GAN) networks, outperforming them in the accuracy, recall, and IoU metrics used.

Tomczak et al. [23] proposed a U-Net model to estimate nodule abundance in seabed images. First, the U-Net segments the nodules and then Connected Component Analysis (CCA) is applied to isolate connected areas on the image. Since the CCA algorithm does not perform well in cases where different nodules form a single region, a watershed algorithm is applied to identify and delineate the boundaries.

Song et al. [24] introduced an improved version of a U-Net to perform image segmentation in mineral images. Their proposal applies an encoder–decoder structure to the U-Net, in which the decoder up-samples the features at several scales to generate the final segmentation map. While the method demonstrated reduced training loss compared to the original U-Net, the limited dataset of 49 seabed mineral images may impact the model's generalisation capabilities.

Besides these deep learning techniques, several approaches have used traditional methods, such as threshold [25,26] and clustering segmentation [27]. Among those using these techniques, Schoening et al. [28] proposed the Compact-Morphology-based poly-metallic Nodule Delineation (CoMoNoD) algorithm. Their algorithm is divided into two steps: contrast maximisation, which is used to generate a binary image, increasing the contrast between the nodules and the seabed sediments, and nodule delineation using several blob detections, splitting, and fusion in order to separate individual nodules. The authors assumed that the nodules have an elliptical and convex shape, and then the potential nodule blobs were outlined by a convex hull. However, the method did not perform well when marine fauna was present in the image and misidentified it as nodules. Hence, deep learning models may

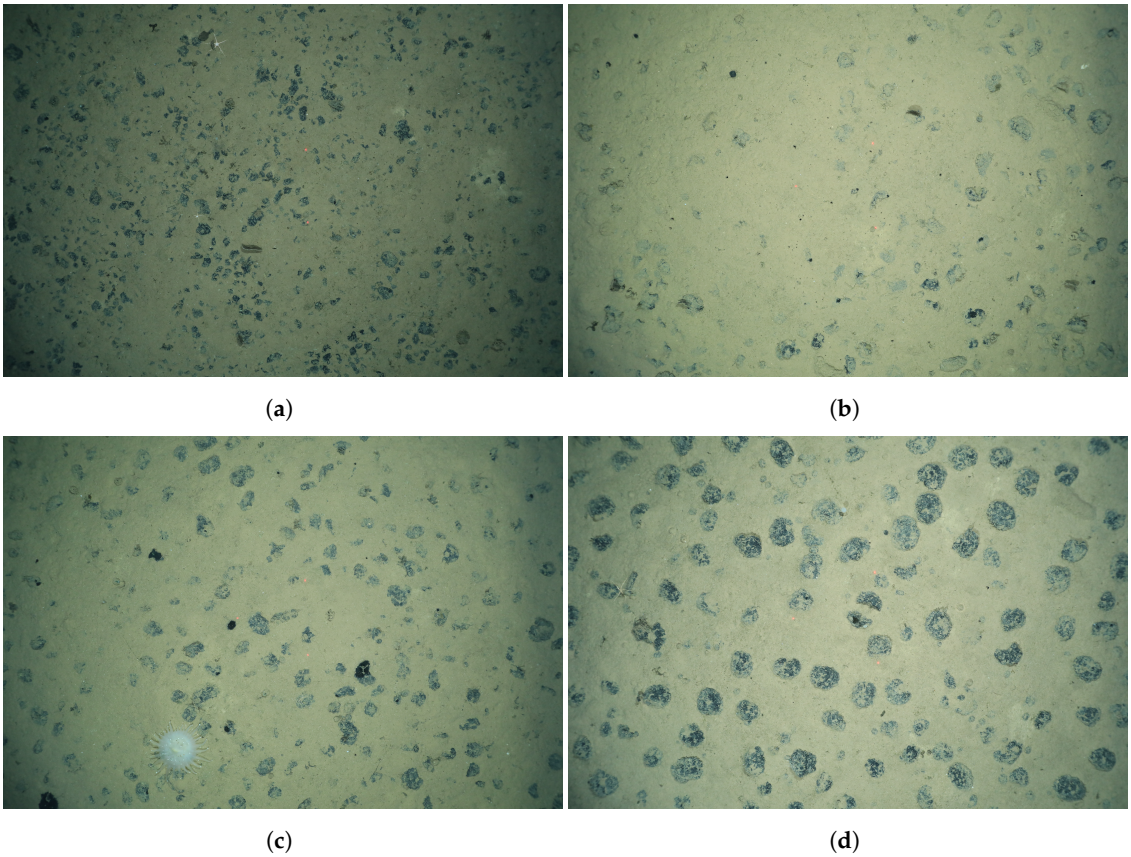
perform better because they can handle complex object shapes and appearance variations, especially in the context of polymetallic nodules, which may be spherical, discoidal, irregular, poly-nodule, or otherwise variably shaped [29]. For instance, binary segmentation may fail when the nodules are partially occluded or buried.

3. Dataset Overview

3.1. Polymetallic Nodule Dataset

This study makes use of an existing dataset that is publicly available and licensed under a Creative Commons Attribution 4.0 International (CC-BY-4.0) license, allowing for its reuse with proper attribution. This dataset, "Seafloor images of undisturbed and disturbed polymetallic nodule province seafloor collected during RV SONNE expeditions SO268/1+2", developed by Purser et al. [30], was collected during the RV SONNE expeditions SO268/1+2 and is composed of raw high-resolution images from the Clarion–Clipperton Fracture Zone (CCZ) in the Pacific Ocean, part of JPI Oceans project MiningImpact [30]. The data collection period spanned from 17 February 2019 to 27 May 2019, ensuring environmental variation, and occurred within the German (BGR) and Belgian (GSR) contract areas [31]. There are 12 subsets that contain 41,088 images in total. The images present diverse lighting, contrast, and brightness conditions, making them ideal for training neural network models. Figure 1 presents example images from the dataset.

The data were acquired using the Ocean Floor Observation System (OFOS), a towed underwater camera system equipped with a high-resolution photo camera (iSiTEC, CANON EOS 5D Mark III) and a high-definition video camera (iSiTEC, Sony FCB-H11). The OFOS setup included two strobe lights, three laser pointers spaced 50 cm apart for scale estimation, four LED lights, a Tritech Altimeter, and a USBL positioning system (Posidonia) for tracking during deployments.



(**a**) (**b**)

(**c**) (**d**)

Figure 1. Original images from the dataset "Seafloor images of undisturbed and disturbed polymetallic nodule province seafloor collected during RV SONNE expeditions SO268/1+2" (images provided by

Purser et al. [30]). (**a**) Image captured in the CCZ at −117.021376 longitude and 11.930071 latitude. (**b**) Image captured in the CCZ at −117.0118958 longitude and 11.86299783 latitude. (**c**) Image captured in the CCZ at −117.0125630 longitude and 11.8621353 latitude. (**d**) Image captured in the CCZ at −125.9254465 longitude and 14.02936067 latitude.

3.2. Annotation Process

Due to the high resolution of the images and the typical architectures of deep learning models, using the entire images for training is not the optimal approach. Resizing the images to smaller sizes can lead to a loss of critical nodule features, while using the full-resolution image substantially increases computational requirements, such as memory usage and training time. Additionally, manual annotation of the entire dataset would be highly time-consuming, considering that one image may have hundreds of nodules. In order to address these challenges, a semi-automated iterative approach was employed, which was as follows:

1. The original images were cropped into smaller patches of 640 × 640 pixels using a sliding window approach. Horizontal and vertical strides of 320 pixels were used to create enough overlap between patches, ensuring that nodules near the edges of the image were included. Zero-padding was added to maintain consistent dimensions. This approach standardized input sizes to enable efficient training across all models. Figure 2 demonstrates the cropping process.

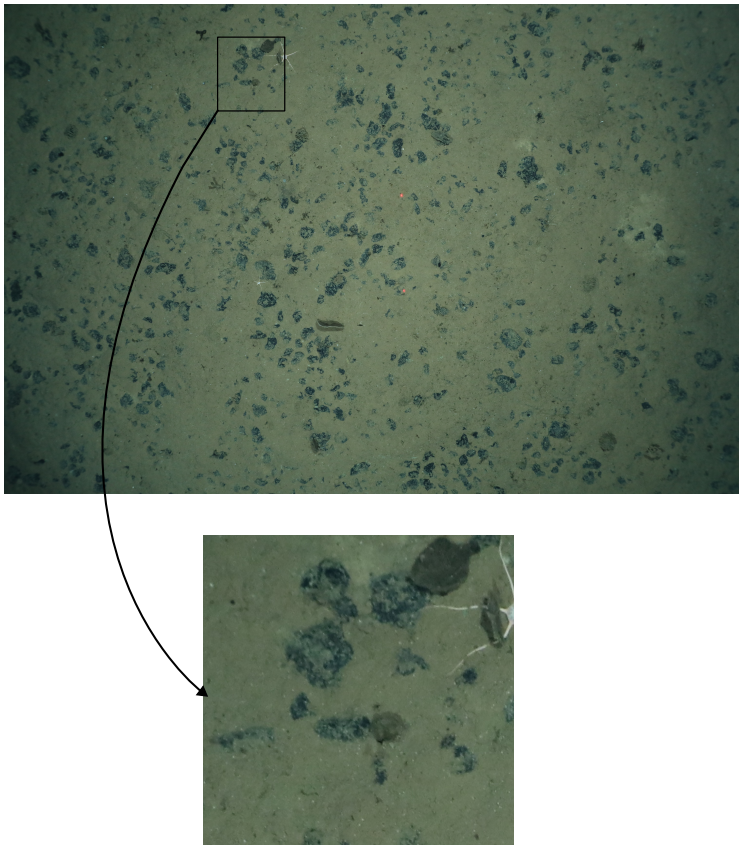


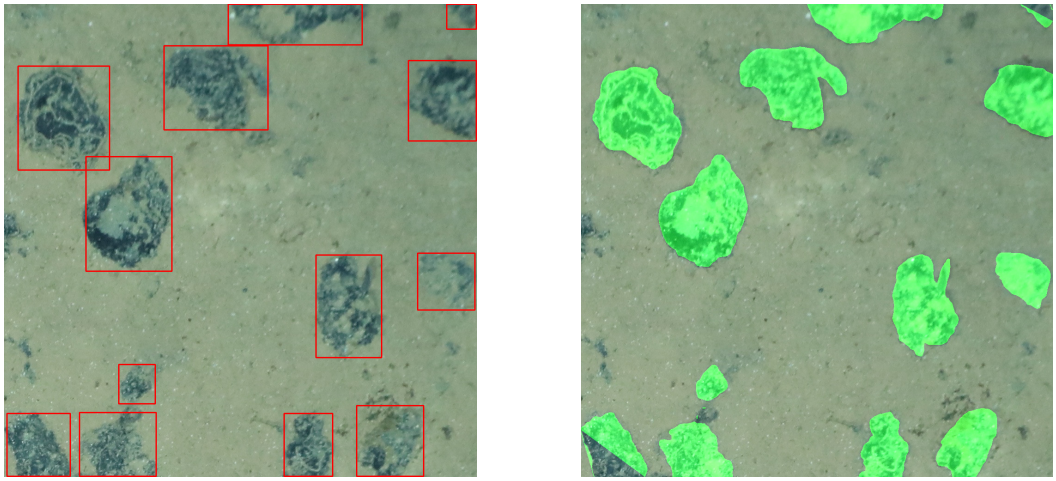
Figure 2. Example of cropped image and original dataset image.

2. The next step consisted of selecting a small subset of cropped patches and manually annotating their nodules using the Supervisely platform [32]. These labels served as the ground truth for training an initial Faster R-CNN model for object detection.
3. The trained model was used to infer and label potential nodules on new images. Then, the predicted bounding boxes were reviewed and corrected, addressing issues such as false negatives and positives. This approach sped up the annotation process.

4. Finally, the corrected labels were added to the dataset, and the model was retrained to improve its performance. This iterative process continued until the model achieved a mean Average Precision (mAP) of 80%. This threshold value minimized the need for manual corrections.

For the segmentation annotation process, the bounding boxes from object detection were used to generate instance masks. The conversion from bounding boxes to instance masks was achieved using the Segment Anything Model (SAM) [33], a state-of-the-art model used for segmentation tasks. The model produced pixel-level masks corresponding to each detected object instance, enabling the delineation of each object's boundaries.

The masks generated by SAM were then reviewed and manually corrected to address potential issues such as over-segmentation, under-segmentation, or inaccuracies in mask boundaries caused by occlusions or overlapping regions. Figure 3 displays the result of converting the bounding box annotations to masks. It is worth noting that some parts of the nodules were not entirely segmented and required manual revision.



(**a**) (**b**)

Figure 3. Conversion from bounding box labels to instance segmentation labels using SAM. (**a**) Bounding box annotations. (**b**) Instance segmentation annotations.

Finally, the semantic segmentation annotation process was carried out by converting the instance masks to a single label and the remaining pixels to a background class.

3.3. Dataset Statistics

Finally, the total dataset used consisted of 779 images with a resolution of 640 × 640 pixels. The dataset was randomly split into three subsets: 70% for training, 20% for validation, and 10% for testing, distributed as in Table 1.

Table 1. Dataset statistics.

Subset	Image Count	Nodule Count
train	546	9255
val	156	2718
test	77	1403
total	779	13,376

4. Benchmark Setup

4.1. Model Selection

Five models were selected and trained for each task: object detection, instance segmentation, and semantic segmentation. Based on related work, we established that state-of-the-art models such as Faster R-CNN, YOLO and Mask R-CNN, U-Net are widely used for detecting or segmenting nodules. For this reason, more recent models were also chosen, such as those based on transformers. The objective was to compare the performance of models based on different architectures.

4.1.1. Object Detection Models

Table 2 summarises the key ideas of the five models chosen for the assessment of their object detection performance.

Table 2. Summary of the evaluated object detection models.

Model	Category Type	Architecture	Key Ideas
Faster R-CNN [34]	Convolutional	Two-stage	– A Region Proposal Network (RPN) is used to generate candidate regions that could potentially contain objects – RoI (Region of Interest) Pooling is applied to extract fixed-size feature representations for each proposal
YOLOv8 [35]	Convolutional	Single-stage	– Directly predicts bounding boxes from the image (anchor-free) – Employs a convolutional backbone and multi-scale feature pyramids
DEtection TRansformer (DETR) [36]	Transformer	Single-stage (transformer encoder–decoder)	– The encoder processes feature maps to learn contextual information – The decoder uses fixed-object queries to focus on relevant features from the encoder's output – The decoder then predicts bounding boxes for detected objects
DETR with Improved deNoising anchOr boxes (DINO) [37]	Transformer	Single-stage (transformer encoder–decoder)	– DINO extends DETR by introducing mixed query denoising, query selection, and a look-forward scheme for box prediction – During training, noise is added to the decoder, and the model learns to reconstruct both the ground truth and noisy queries
EfficientNet [38]	Convolutional	Single-stage	– The model uses a compound scaling method to balance depth, width, and resolution for multi-scale feature extraction – A Bi-Directional Feature Pyramid Network (BiFPN) enables adaptive multi-scale feature fusion for improved detection across object sizes – The model employs anchor boxes for bounding box predictions

4.1.2. Instance Segmentation

In addition to object detection, a YOLOv8 model was trained for instance segmentation. The other four models are summarised in Table 3.

Table 3. Summary of the evaluated instance segmentation models.

Model	Category Type	Architecture	Key Ideas
Mask R-CNN [39]	Convolutional	Two-stage	– Extends Faster R-CNN by adding a branch for pixel-level mask prediction – Uses RoIAlign to align extracted features with input images accurately – The model adds a segmentation mask for each detected object
Segmenting Objects by Locations (SOLOv2) [40]	Convolutional	Single-stage	– Directly predicts instance masks in a single forward pass without an intermediate object detection stage – The model divides the image into a uniform grid, predicting instance masks for each cell
Point-based Rendering (PointRend) [41]	Convolutional	Two-stage	– Two-stage instance and semantic segmentation method – The method iteratively refines predictions at selected points near the edges instead of directly predicting all masks simultaneously
Mask2Former [42]	Transformer	Multi-stage (transformer encoder–decoder)	– The model predicts a fixed set of masks and their associated categories using attention mechanisms – The model employs query embeddings, and each query represents a specific region – The decoder iteratively refines the queries to predict the masks

4.1.3. Semantic Segmentation

Besides PointRend, the other four models chosen for the semantic segmentation task are presented in Table 4:

Table 4. Summary of the evaluated semantic segmentation models.

Model	Category Type	Architecture	Key Ideas
U-Net [43]	Convolutional	Fully convolutional (encoder–decoder)	– Symmetric encoder–decoder architecture – The encoder conducts several convolutional and down-sampling operations to obtain high-level features – The decoder uses up-sampling layers to reconstruct a dense pixel-wise output
DeepLabv3+ [44]	Convolutional	Encoder–decoder with Atrous Convolutions	– Encoder–decoder architecture used to gather multi-scale features based on Atrous Spatial Pyramid Pooling (ASPP) – The decoder refines the segmentation output by recovering spatial details through up-sampling and feature fusion
MobileNetv3 [45]	Convolutional	Encoder–decoder	– Two-stage instance and semantic segmentation method – Optimised for mobile and edge devices – Used as an encoder for semantic segmentation tasks – Lite Reduced Atrous Spatial Pyramid Pooling (LRASPP) Head as the decoder – The decoder applies convolutions for multi-scale context aggregation, generating dense segmentation maps
SegFormer [46]	Transformer	Hierarchical Transformer Backbone	– The model has a lightweight design and uses hierarchical transformers as a backbone to extract global and local information

4.2. Training Setup

In order to ensure consistency during training, the hardware setup used to train and evaluate each task was the same. All experiments were conducted using an NVIDIA GeForce GTX 1080 Ti GPU with 12 GB VRAM, an Intel Core i9-10940X @ 3.30 GHz (28 cores) CPU, and 128 GB of RAM. The training was carried out using Ubuntu 20.04.6 as the operating system. The MMDetection and MMSegmentation frameworks [47,48] were utilised for implementing the models. These frameworks are open-source object detection and segmentation toolboxes based on PyTorch, which enables the seamless configuration of state-of-the-art architectures.

The hyperparameters used for training were chosen to balance computational efficiency and performance. The hyperparameter configuration was applied uniformly across all models when possible, ensuring a consistent comparison of their performance. Due to GPU limitations, a batch size of 2 was used. Besides this, the models were trained for 12 epochs, a value previously selected to allow for sufficient convergence while avoiding overfitting. As exceptions, the YOLOs models were trained with a batch size of 8 and 100 epochs due to their lightweight architecture and the DINO and DETR models were trained with a batch size of 1 due to GPU restraints.

The learning rate schedule followed a multi-step decay policy, where the learning rate was reduced by 10% at epochs 4 and 8. This approach hastened initial learning and stabilised training in later stages, enabling the models to converge effectively. The optimiser chosen varied depending on the model, with Stochastic Gradient Descent (SGD [49]) and AdamW [50] being chosen. Table 5 summarises the hyperparameter setup.

Finally, we have utilised pre-trained models provided by the MMDetection and MMSegmentation toolboxes to train the new models. By using transfer learning, the need for extensive data and computational resources was reduced. Given their larger GPU requirements, the batch size for the DINO and DETR models was set to 1.

Table 5. Hyperparameter setup for each model.

Task	Model	Batch Size	Epochs	Optimiser	Learning Rate Schedule
Object Detection	YOLOv8	8	100	AdamW	Initially 0.01, reduce 10% at epochs 4, 8
	Faster R-CNN	2	12	SGD	Initially 0.01, reduce 10% at epochs 4, 8
	DINO	1	12	AdamW	Initially 0.01, reduce 10% at epochs 4, 8
	EfficientDet	2	12	SGD	Initially 0.01, reduce 10% at epochs 4, 8
	DETR	1	12	AdamW	Initially 0.01, reduce 10% at epochs 4, 8
Instance Segmentation	YOLOv8s-seg	8	100	AdamW	Initially 0.01, reduce 10% at epochs 4, 8
	Mask R-CNN	2	12	SGD	Initially 0.01, reduce 10% at epochs 4, 8
	PointRend	2	12	SGD	Initially 0.01, reduce 10% at epochs 4, 8
	SOLOv2	2	12	SGD	Initially 0.01, reduce 10% at epochs 4, 8
	Mask2Former	2	12	AdamW	Initially 0.01, reduce 10% at epochs 4, 8
Semantic Segmentation	U-Net	2	12	Adam	Initially 0.01, reduce 10% at epochs 4, 8
	DeepLabV3+	2	12	Adam	Initially 0.01, reduce 10% at epochs 4, 8
	SegFormer	2	12	AdamW	Initially 0.01, reduce 10% at epochs 4, 8
	PointRend	2	12	Adam	Initially 0.01, reduce 10% at epochs 4, 8
	MobileNetv3	2	12	Adam	Multi-step: reduce 10% at epochs 4, 8

Additionally, data augmentation techniques were implemented during training to increase the models' generalization and robustness. Transformations such as random flips, rotations, scale changes, and changes in brightness and contrast were applied to the training data.

4.3. Evaluation Metrics

Standard metrics were chosen to guarantee an efficient evaluation of the models' performance of each task. The standard COCO evaluation metrics were used for object detection and instance segmentation. These metrics permit us to assess the models' performance across varying levels of overlap.

- Mean Average Precision (mAP): mAP is calculated as the mean of the average precision (AP) over multiple Intersection over Union (IoU) thresholds (0.5 to 0.95 in steps of 0.05) [51].
  For a specific IoU threshold t, AP is defined as

$$\mathrm{AP}(t) = \frac{1}{N} \sum_{i=1}^{N} P(i) \Delta R(i), \tag{1}$$

  where N, $P(i)$, $R(i)$, and $\Delta R(i)$, are the total number of object classes. Precision is determined at the i-th threshold, and recall and change in recall at i.

Then, the mAP is computed as follows:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N}\text{AP}(i) \tag{2}$$

For a better understanding of the models' accuracy, we evaluated their performance over different IoU, using mAP_{50} and mAP_{75}. Besides this, we also assessed their performance at different object scales based on pixel size, using the mAP_{s}, mAP_{m}, and mAP_{l} metrics, which corresponded to small (less than 32×32 pixels), medium (32×32 to 96×96), and large objects (greater than 96×96).

On the other hand, the semantic segmentation models were evaluated using metrics that require pixel-wise classification, measuring the overlap between the predicted and ground-truth areas.

- Accuracy: The accuracy metric evaluates the correctly predicted pixels in comparison to the total number of pixels [52]. This metric may be inadequate for use in cases of class imbalance since the favoured class increases the accuracy score.
- Intersection over Union: This quantifies the overlap between predictions and ground truth [53] and is computed as follows:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{3}$$

The mean Intersection over Union (mIoU) is also used to measure the mean IoU over the nodules and background classes. The mIoU is computed considering a target n.

- Dice Coefficient: A harmonic mean of precision and recall [54], which is computed as follows:

$$\text{Dice} = \frac{2 \cdot \text{Area of Overlap}}{\text{Total Area of Predicted and Ground Truth}} \tag{4}$$

Similarly, mDice is also utilised to measure the mean over all targets.

All metrics are complementary in assessing segmentation accuracy, penalising false positives.

5. Results

The models were implemented using three established frameworks. Both YOLOv8 models were implemented using the Ultralytics repository [55]. The remaining object detection and instance segmentation models were trained using the MMDetection toolbox, while the MMSegmentation toolbox was used for semantic segmentation. These tools enable seamless implementation, providing robust and modular architectures for training and evaluation.

Since the MMDetection and MMSegmentation toolboxes do not compute validation losses, we compared the models' map trends on the validation set with their training losses to identify potential overfitting. Additionally, an early stopping mechanism was employed to avoid overfitting.

5.1. Object Detection Results

Table 6 summarises the evaluation metrics obtained for all models. DINO demonstrated the highest overall performance, achieving a mAP of 0.899 and outperforming all other models across all evaluated metrics. These results highlight the model's ability to detect objects over several object scales and IoU thresholds. YOLOv8s and Faster R-CNN achieved reasonable results, although inferior to those of DINO. In contrast, DETR and EfficientNet showed relatively poor performances, indicating a need for additional training epochs.

Table 6. Evaluation metrics for object detection. Best results are presented in bold.

Model	mAP	mAP_50	mAP_75	mAP_s	mAP_m	mAP_l
YOLOv8s	0.856	0.974	0.955	0.295	0.840	0.878
Faster R-CNN	0.832	0.979	0.957	0.227	0.814	0.851
DINO	**0.899**	**0.990**	**0.979**	**0.472**	**0.895**	**0.910**
DETR	0.772	0.967	0.928	0.256	0.752	0.800
EfficientNet	0.7473	0.936	0.904	0.375	0.731	0.759

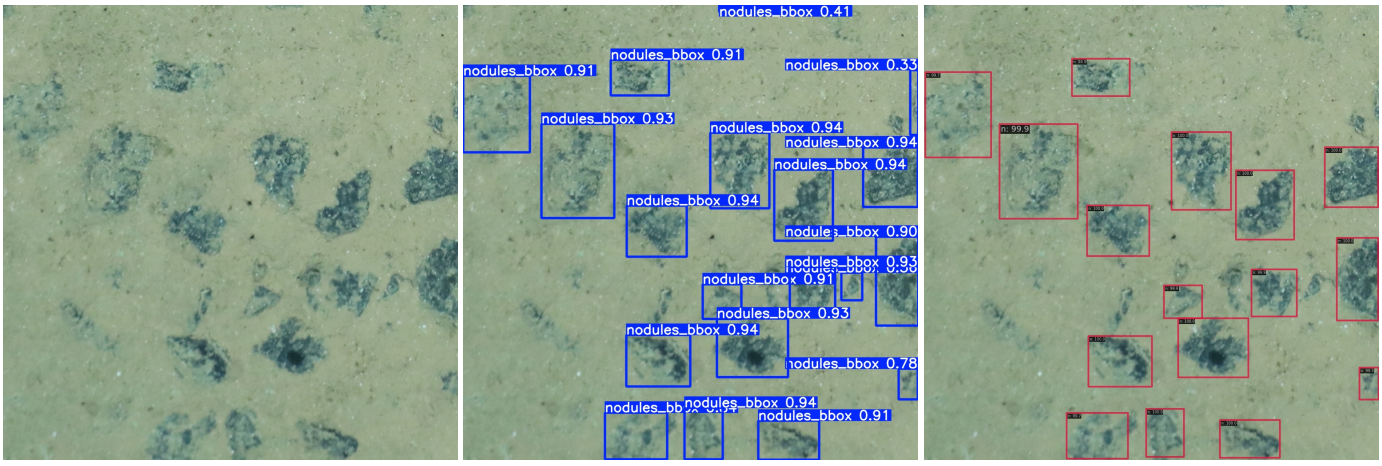
The majority of the models struggled to detect small objects. Considering the deep-sea mining scenario, in which nodules can be partially buried or fragmented into less distinct clusters, their low performance in this metric is a challenge for DL models.

The models were also evaluated in terms of their computational performance to assess their training efficiency and resource usage. Considering that the other models were trained for only 12 epochs and used a smaller batch size, YOLOv8s presented the fastest training time despite its larger workload, attributable to its lightweight design. Additionally, YOLOv8s outperformed the others in inference speed and GPU usage, highlighting its suitability for deployment on edge devices. On the other hand, the transformer-based models were computationally intensive. Among the models trained for 12 epochs, Faster R-CNN achieved the shortest training duration, making it the most time-efficient for smaller training setups. Table 7 displays the computational performance of each object detection model.

Table 7. Computational performance of models in terms of object detection. Best results are presented in bold.

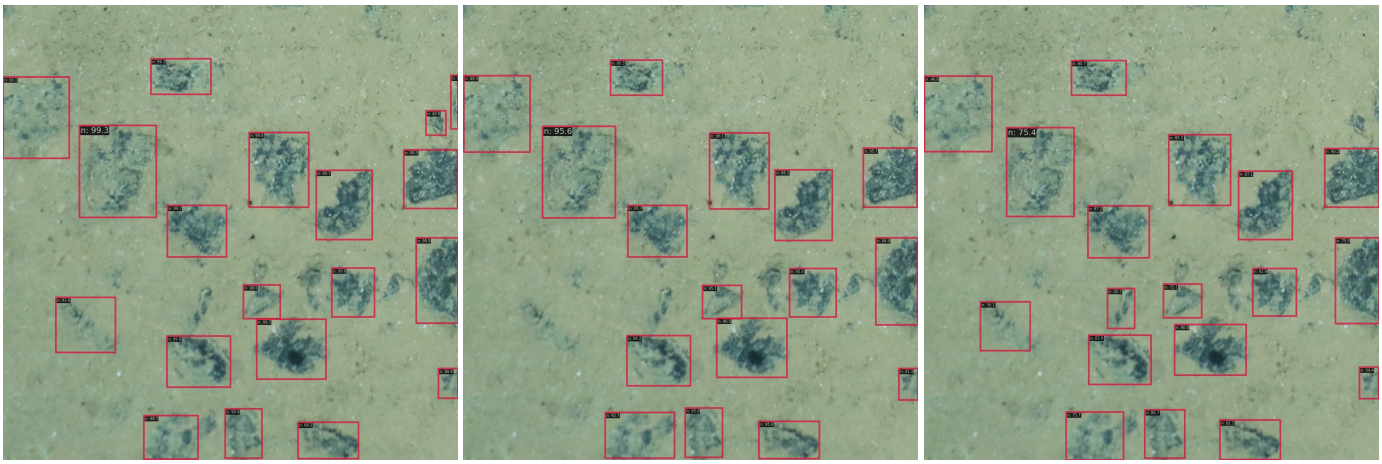
Model	Training Duration	Inference Time per Image	Max GPU Memory Used
YOLOv8s	42 min 41 s	**28.9 ms**	**105.54 MB**
Faster R-CNN	**26 min 26 s**	72.7 ms	423.90 MB
DINO	2 h 11 min 32 s	128.6 ms	419.01 MB
DETR	1 h 20 min 23 s	48.6 ms	396.17 MB
EfficientNet	34 min 39 s	87.1 ms	428.84 MB

To illustrate the results obtained, Figure 4 shows the bounding boxes created for each detection. It is worth noting that, despite the good values obtained in the evaluation metrics, the models were not able to localise all the nodules.



(**a**) (**b**) (**c**)

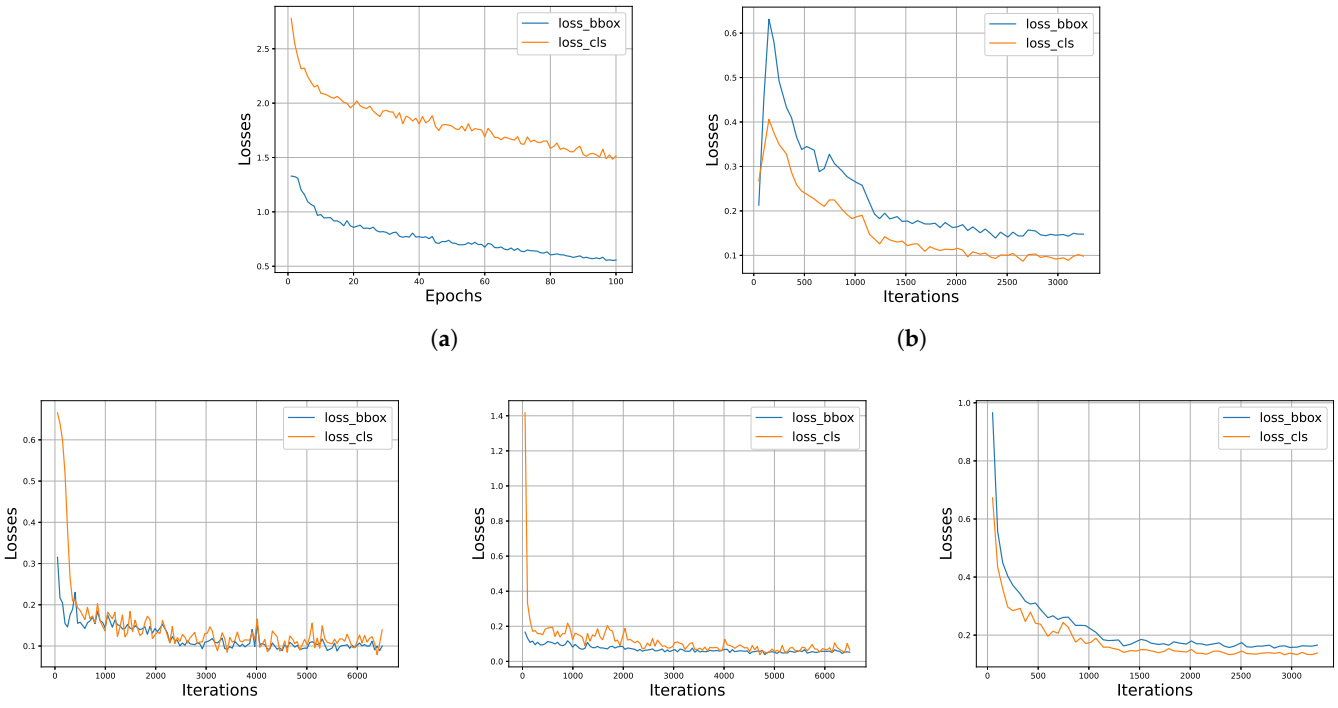
Figure 4. *Cont.*



(**d**) (**e**) (**f**)

Figure 4. Example of object detection results for each model. (**a**) Original image. (**b**) YOLOv8s detection results. (**c**) Faster R-CNN detection results. (**d**) DETR detection results. (**e**) DINO detection results. (**f**) EfficientNet detection results.

Since MMDetection does not compute validation losses, the training losses were also logged to assess possible overfitting during training. Figure 5 displays the localisation (in blue) and classification (in orange) achieved by all models. It is noticeable that all losses tend to decrease rapidly before stabilising, with both losses converging to relatively satisfactory values. This behaviour, coupled with the mAP values obtained for validation, indicates that the models are not overfitting.



(**a**) (**b**)

(**c**) (**d**) (**e**)

Figure 5. Localisation and classification losses for each object detection model. (**a**) YOLOv8s localisation (blue) and classification (orange) losses. (**b**) Faster R-CNN localisation (blue) and classification (orange) losses. (**c**) DETR localisation (blue) and classification (orange) losses. (**d**) DINO localisation (blue) and classification (orange) losses. (**e**) EfficientNet localisation (blue) and classification (orange) losses.

5.2. Instance Segmentation Results

Similarly, the instance segmentation models were evaluated regarding their localization efficiency and ability to identify nodules of different sizes. PointRend presented the highest mAP, followed closely by SOLOv2 and Mask R-CNN. At higher IoU thresholds, SOLOv2 attained better results, suggesting it had a better capacity to localise a larger number of nodules, excelling at capturing spatial and fine-grained features. Considering the scale-based metrics used, YOLOv8 had difficulty identifying small nodules, as expected. Conversely, SOLOv2 scored the best mAP value for small objects. Most models performed well for medium and large objects, with the exception of Mask2Former and YOLOv8-seg. Despite its transformer-based architecture, the poor performance of the Mask2Former model in comparison with the other models may indicate a need for longer training durations and larger batch sizes. Interestingly, the models obtained better results for segmenting small-scale nodules than the object detection models. In the case of YOLOv8, there was a drop in performance compared to the object detection model. Table 8 displays the evaluation metrics for all models.

Table 8. Evaluation metrics for instance segmentation. Best results are presented in bold.

Model	mAP_seg	mAP_50_seg	mAP_75_seg	mAP_s_seg	mAP_m_seg	mAP_l_seg
YOLOv8s-seg	0.493	0.611	0.551	0.175	0.484	0.858
Mask R-CNN	0.842	0.968	0.941	0.538	0.844	**0.967**
PointRend	**0.857**	**0.973**	0.937	0.515	**0.856**	0.900
SOLOv2	0.849	**0.973**	**0.943**	**0.571**	0.846	0.894
Mask2Former	0.523	0.586	0.567	0.001	0.625	0.814

Consistent with the results seen in object detection tasks, YOLOv8 achieved the best overall performance concerning computational requirements, as outlined in Table 9. Conversely, the transformer-based Mask2Former showed the highest inference time, mirroring the behaviour of its object detection counterpart, DINO.

Table 9. Computational performance of models in terms of instance segmentation. Best results are presented in bold.

Model	Training Duration	Inference Time per Image	Max GPU Memory Used
YOLOv8s-seg	56 min 9 s	**12.3 ms**	**202.45 MB**
Mask R-CNN	**40 min 57 s**	94.4 ms	482.09 MB
PointRend	43 min 3 s	117.0 ms	995.41 MB
SOLOv2	59 min	171.1 ms	757.30 MB
Mask2Former	57 min 12 s	223.4 ms	2506.53 MB

Figure 6 presents examples of segmentation achieved by all models. Despite the proximity of some nodules, the models successfully segmented them, achieving relatively high accuracy scores.

Similarly, the mask and classification losses were also obtained for each instance segmentation model, as presented in Figure 7. With the exception of the Mask2Former model, the losses for the remaining models showed the same pattern of a quick drop and stabilisation. The high variance of the Mask2Former model may be due to the small batch size, with its classification loss presenting larger spikes. This behaviour is aligned with the metrics achieved, showing its poor performance compared to the other models.

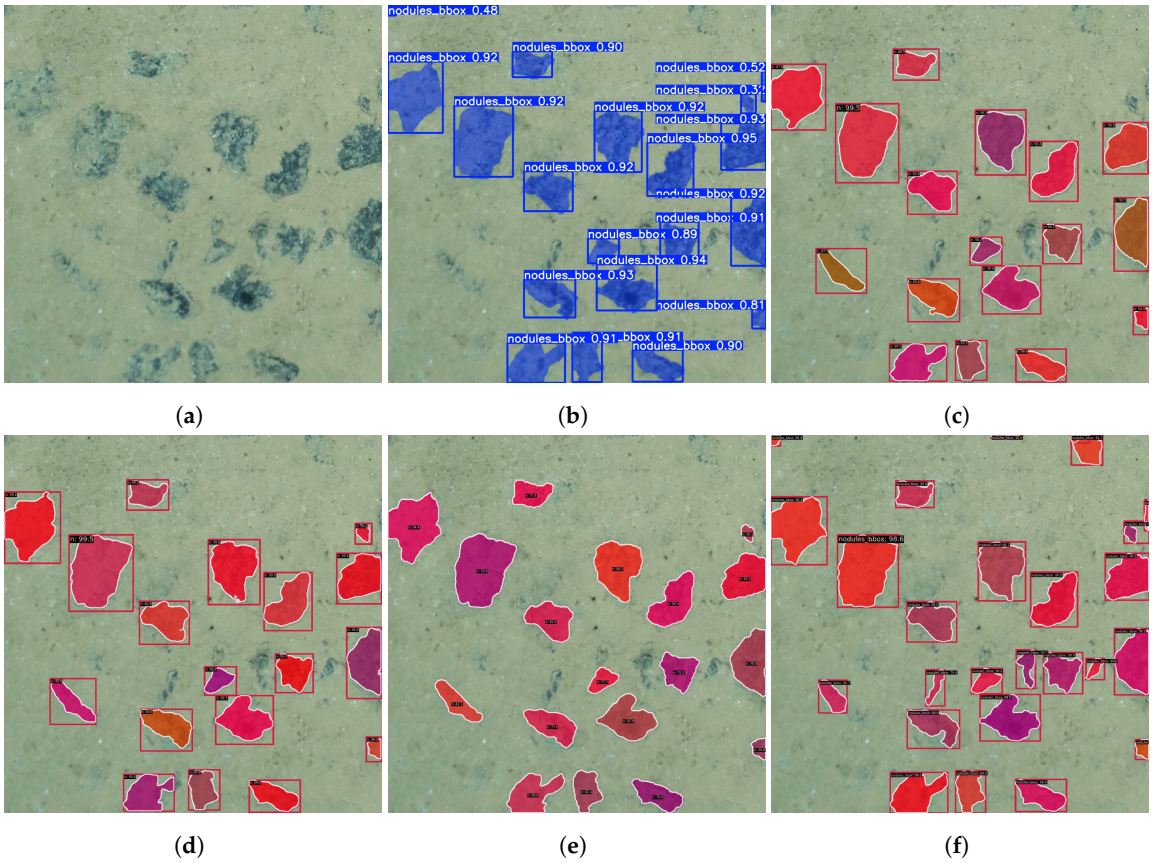


Figure 6. Examples of instance segmentation results for each model. (**a**) Original image. (**b**) YOLOv8s segmentation results.(**c**) Mask R-CNN segmentation results. (**d**) PointRend segmentation results. (**e**) SOLOv2 segmentation results. (**f**) Mask2Former segmentation results.

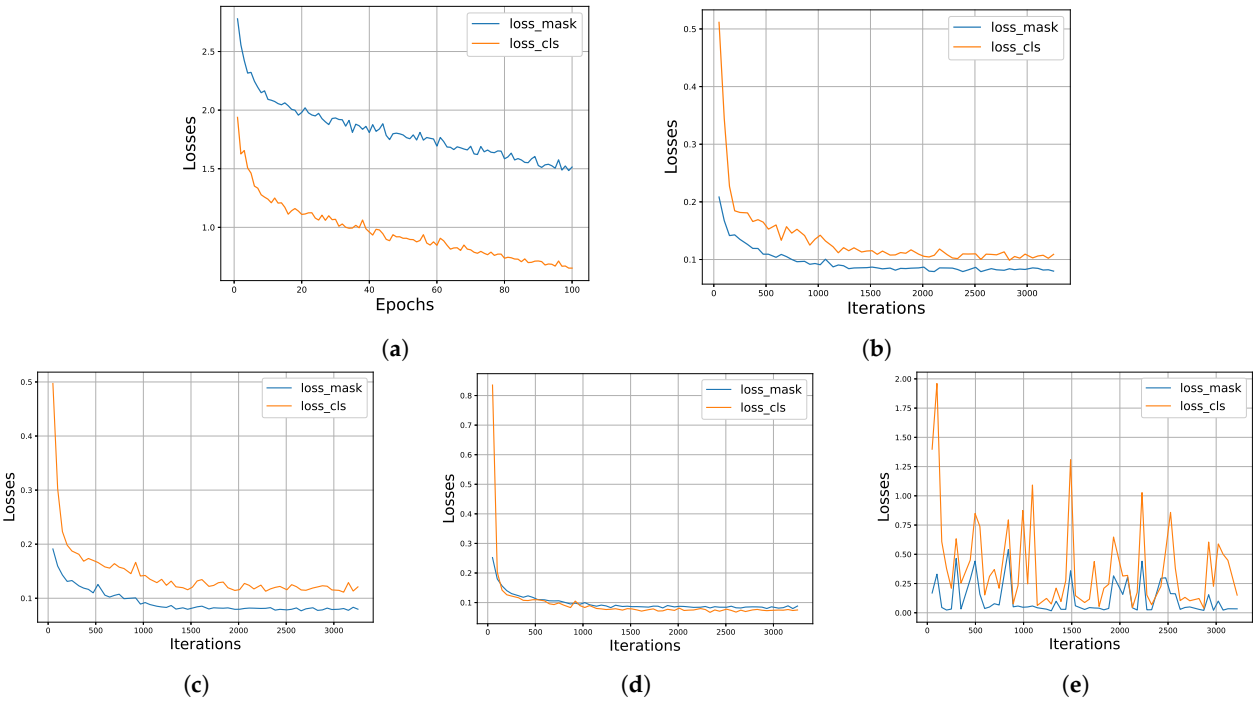


Figure 7. Segmentation and classification losses for each instance segmentation model. (**a**) YOLOv8s mask (in blue) and classification (in orange) losses. (**b**) Mask R-CNN mask (in blue) and classification (in orange) losses. (**c**) PointRend mask (in blue) and classification (in orange) losses. (**d**) SOLOv2 mask (in blue) and classification (in orange) losses. (**e**) Mask2Former mask (in blue) and classification (in orange) losses.

5.3. Semantic Segmentation

PointTrend and Segformer achieved the highest performance overall, with Segformer slightly outperforming the others in terms of mIoU (75.41 and 73.85, respectively) and mDice (84.8 and 83.69, respectively). Both models demonstrated superior capabilities in handling segmentation tasks, achieving an aAcc of over 92%. DeepLabv3+ followed with a moderate performance, while MobileNetV3 achieved a lower mIoU but comparable mAcc to the higher-performing models. Among all models, Unet demonstrated the lowest overall and class-specific performance, struggling particularly with nodule localisation. Table 10 displays the results for semantic segmentation.

Table 10. Evaluation metrics for semantic segmentation. Best results are presented in bold.

Model	aAcc (%)	mIoU (%)	mAcc (%)	mDice (%)
PoinTrend	92.75	73.85	**88.65**	83.69
Segformer	**93.72**	**75.41**	87.0	**84.8**
MobileNetV3	86.22	61.94	83.75	73.97
U-Net	54.3	33.34	68.46	47.76
DeepLabv3+	84.24	59.97	84.98	72.42

Regarding the class-specific analysis, all models have a low IoU for nodules compared to the background, even with a relatively high accuracy. Therefore, this may suggest that the models can correctly identify polymetallic nodule objects but fail to segment their regions. Additionally, given that there are model background pixels rather than nodules, the models favour the background class. Table 11 summarises the results.

Table 11. Class-specific evaluation metrics for semantic segmentation. Best results are presented in bold.

Model	Class	IoU (%)	Acc (%)	Dice (%)
PoinTrend	nodules	55.69	83.42	71.54
	background	92.02	93.89	95.84
Segformer	nodules	**57.69**	78.41	73.17
	background	93.13	95.59	**96.44**
MobileNetV3	nodules	39.0	80.59	56.11
	background	84.89	86.91	91.83
U-Net	nodules	17.15	**86.57**	29.28
	background	49.53	50.34	66.24
DeepLabv3+	nodules	37.34	85.93	54.37
	background	82.61	84.03	90.48

Regarding computational performance, the lightweight model MobileNetV3 attained the fastest inference time and smallest GPU usage, as displayed in Table 12. This result was expected, as the model balances efficiency and speed. On the other hand, Unet displayed the longest training time and the slowest inference speed; hence, it is less suitable for real-time scenarios.

Figure 8 illustrates the results for semantic segmentation. It is worth noting that certain models, such as DeepLabv3+, struggled to distinguish adjacent nodule regions, grouping them into one region. PointRend and UNet achieved better performances in identifying the small-sized nodules in the images.

Table 12. Computational performance of models in terms of semantic segmentation. Best results are presented in bold.

Model	Training Duration	Inference Time per Image	Max GPU Memory Used
PointRend	**21 min 15 s**	249.1 ms	1848.49 MB
Segformer	1 h 12 min 29 s	362.3 ms	1440.40 MB
MobileNetV3	31 min 2 s	**152.9 ms**	**1300.24 MB**
U-Net	3 h 52 min 8 s	6090.5 ms	2092.89 MB
DeepLabv3+	1 h 13 min 56 s	547.5 ms	4895.89 MB

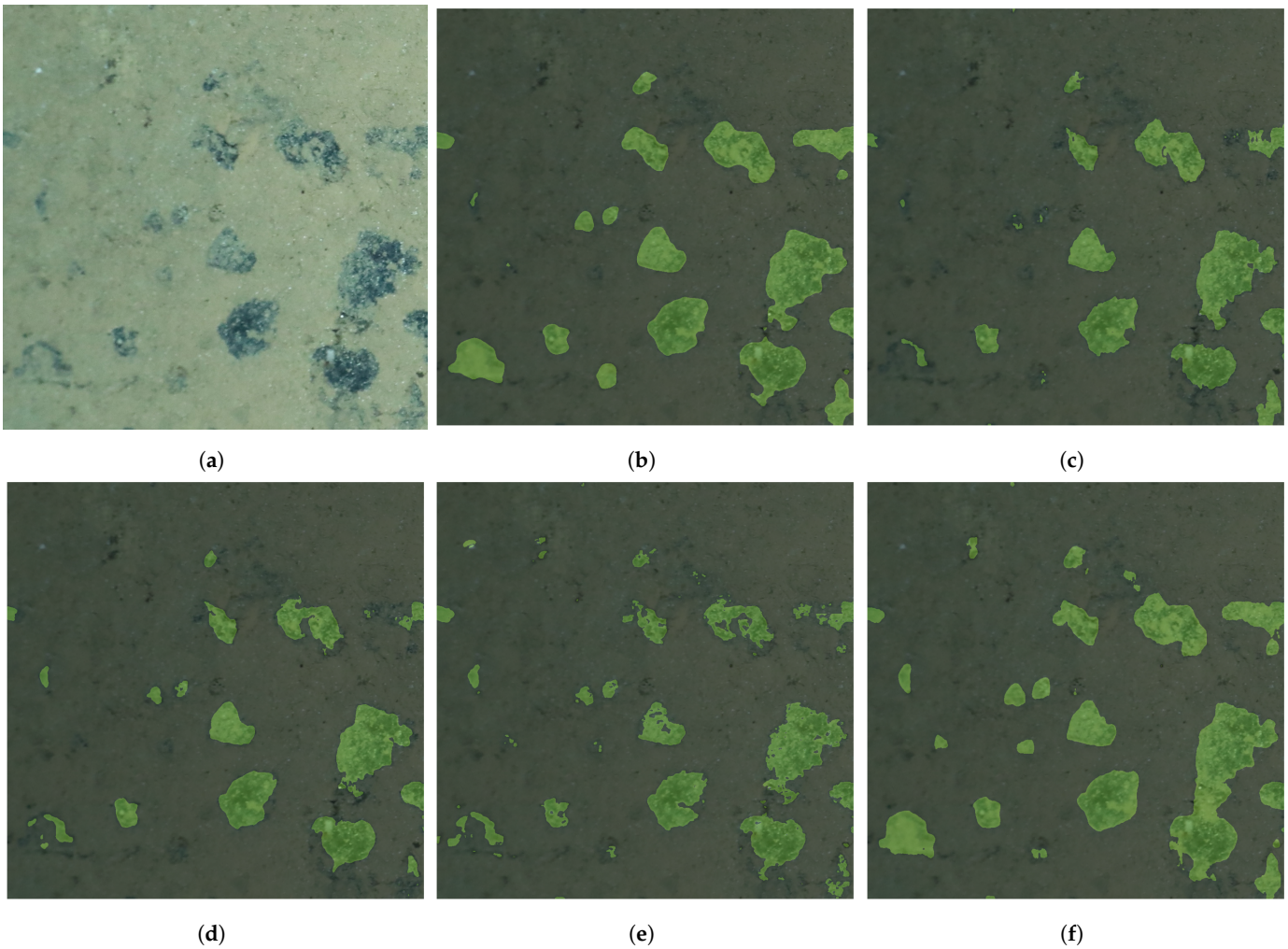


Figure 8. Examples of semantic segmentations results for each model. (**a**) Original image. (**b**) PointRend segmentation results. (**c**) Segformer segmentation results. (**d**) MobilenNetV3 segmentation results. (**e**) U-Net segmentation results. (**f**) DeepLabv3+ segmentation results.

Figure 9 displays the segmentation loss trends for all models during training. All models presented a downward trend, indicating effective learning and convergence. However, the variations seen at the end of training suggest that the models do not fully stabilise, indicating that the number of epochs may need to be increased.

5.4. Colour Analysis

Due to the great variety in the shapes and contours of the nodules, we decided to analyse the effect of colour as a feature, as there was a large contrast between the nodules and surrounding sediments. To accomplish this, the test-subset images were converted

to greyscale and the models were evaluated again. Table 13 displays the results for the greyscale images. All models showed a sharp drop in performance when tested on greyscale images, with only the YOLOv8s object detection model achieving better results. These results suggest that the models rely on features presented in colour channels. Surprisingly, the Mask2Former model achieved the best results in greyscale segmentation, which might indicate a strong reliance on structural and spatial information.

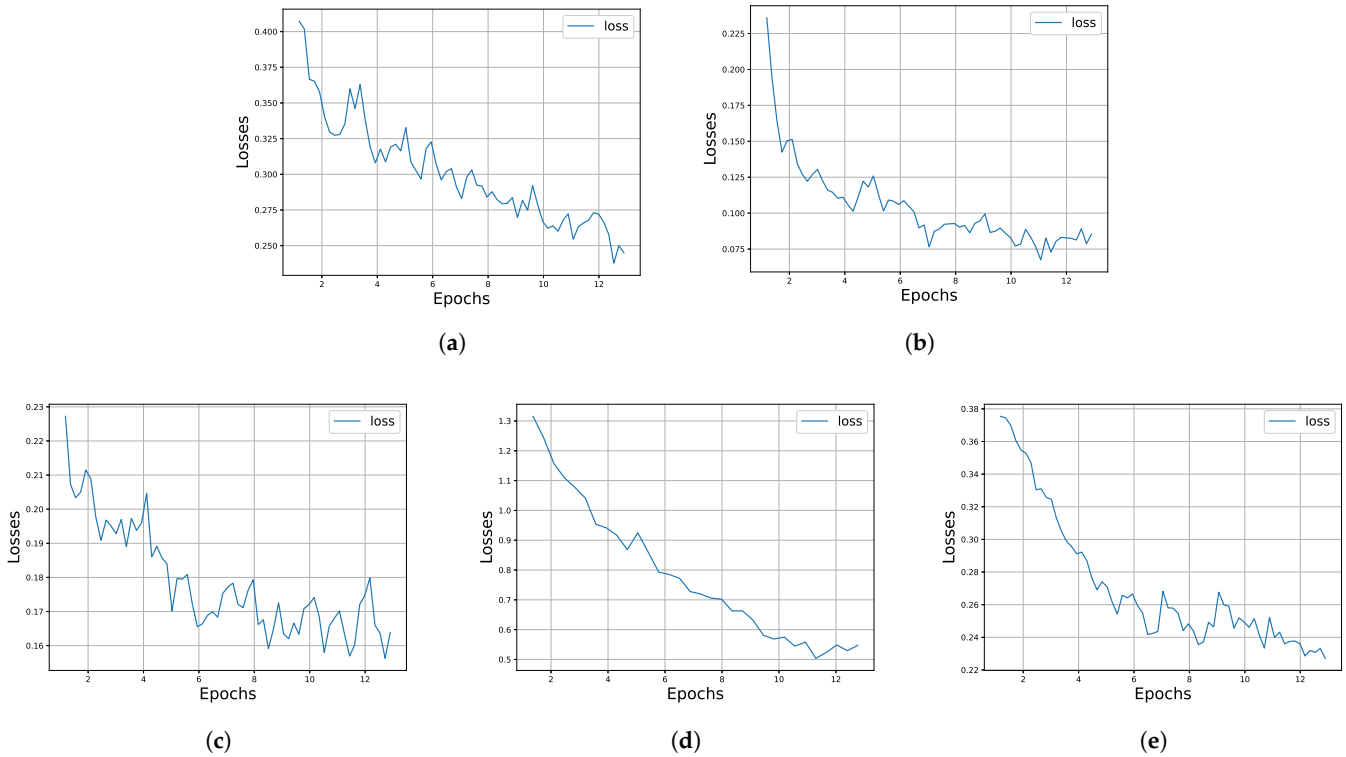


Figure 9. Segmentation loss of each model. (**a**) PointRend segmentation loss. (**b**) Segformer segmentation loss. (**c**) MobilenNetV3 segmentation loss. (**d**) U-Net segmentation loss. (**e**) DeepLabv3+ segmentation loss.

Table 13. Evaluation metrics for greyscale images.

Detection Models	**mAP**	**mAP_50**	**mAP_75**
YOLOv8s-detection	0.591	0.830	0.755
Faster R-CNN	0.335	0.529	0.401
DINO	0.449	0.718	0.557
DETR	0.481	0.777	0.558
EfficientNet	0.498	0.790	0.543
Segmentation Models	**mAP**	**mAP_50**	**mAP_75**
YOLOv8s-seg	0.343	0.479	0.398
Mask R-CNN	0.484	0.601	0.554
PointRend	0.547	0.692	0.624
SOLOv2	0.564	0.722	0.649
Mask2Former	0.576	0.767	0.667

6. Discussion

Novel transformer-based methods, such as DINO for object detection and Segformer for semantic segmentation, demonstrate better results due to their ability to effectively

model global contexts and capture fine-grained details. Unlike traditional convolutional architectures, transformers rely on self-attention mechanisms that enable them to simultaneously process the relationships between all pixels or features, making them particularly well-suited for complex tasks like detecting and segmenting polymetallic nodules in cluttered or noisy environments. However, Mask2Former showed worse results on the instance segmentation task. This may indicate the need to fine-tune the hyperparameters to improve this model's behaviour, such as increasing the batch size and number of epochs.

The results show a trade-off between speed and performance, particularly when comparing lightweight architectures like YOLOv8s to more computationally intensive transformer-based models. Considering that YOLOv8 is designed for real-time application, the results highlight its applicability for AUVs or ROVs. Similarly, while Segformer achieved a better performance in semantic segmentation, it needs more processing power than simpler models like MobileNetV3. This trade-off emphasises the need to balance task requirements, where the choice of model may depend on the need for greater accuracy or hardware constraints.

In addition, identifying small nodules and delineating their boundaries pose significant challenges for the evaluated models, as reflected in their lower performance on small objects and the nodules' class-specific metrics. These limitations are probably due to the irregular morphology of nodules, their visual similarity to the surrounding seabed, and insufficient boundary refinement.

The results indicate a class imbalance issue within semantic segmentation due to the bias toward the background class. Since most image pixels correspond to the background, the networks fail to learn features in the training data. Thus, further strategies must be applied to improve their performance, such as using weighted loss functions.

Finally, the greyscale test indicates that the models rely heavily on the colour channels. Due to the varying shapes and formats of the nodules, the models struggle to generalise. One potential solution is to include greyscale images in the training set and apply contrast-based data augmentation during training.

Although the results are relatively satisfactory, there remains potential for improvement. One potential improvement could be the use of multimodal learning. Prior studies in the literature have used deep learning approaches to characterise polymetallic nodules in acoustic data (e.g., MBES backscatter or side-scan sonar) [56,57]. Thus, developing a method for fusing optical and acoustic features could have a significant impact on the polymetallic nodules' characterisation. For instance, backscatter data can be used to obtain depth information, which, when fused with optical data, may enable the estimation of 3D bounding boxes.

7. Conclusions

This paper evaluated the performance of deep learning models in three different tasks: object detection, instance segmentation, and semantic segmentation. To this end, a semi-autonomous approach was adopted to annotate the dataset. Additionally, the logic for choosing the trained models was based on comparing classic architectures and more recent methods, such as those based on transformers.

The results indicate that the transformer-based techniques perform very efficiently in terms of precision. However, this improvement requires larger computational resources. Considering that deep sea mining uses autonomous vehicles such as ROVs and AUVs, which generally do not have large GPU capacities, these techniques should be chosen according to the hardware setup. In this context, lighter models such as YOLO showed results that were very competitive with those of the transformer-based methods, presenting good accuracy and faster inference.

Nevertheless, there is significant room for improvement. Further studies are required to thoroughly evaluate the performance of these models, particularly their generalisability to other images. Furthermore, the presence of marine life still needs to be evaluated. While some images reveal instances of marine fauna, their occurrence is significantly less frequent than that of the nodules, leading to a notable class imbalance. Future work should focus on adding marine life classes to the dataset to increase variety, improving the models' generalisation capabilities.

Author Contributions: Conceptualization, G.L.; methodology, G.L.; software, G.L.; validation, G.L.; formal analysis, G.L.; investigation, G.L.; resources, E.S.; data curation, G.L.; writing—original draft preparation, G.L.; writing—review and editing, G.L., A.D., J.A., A.M. and E.S.; visualization, G.L.; supervision, A.D., J.A., A.M. and E.S.; project administration, E.S.; funding acquisition, E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by TRIDENT project financed by the European Union's HE programme under grant agreement No 101091959, by National Funds through the Portuguese funding agency, FCT—Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020 (DOI 10.54499/LA/P/0063/2020), and by FCT for the Ph.D. Grant 2021.08715.BD.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This work makes use of the existing dataset "Seafloor images of undisturbed and disturbed polymetallic nodule province seafloor collected during RV SONNE expeditions SO268/1+2", which can be downloaded from https://doi.pangaea.de/10.1594/PANGAEA.935856 [30], accessed on 8 January 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AUV	Autonomous Underwater Vehicle
BiFPN	Bi-Directional Feature Pyramid Network
CCA	Connected Component Analysis
CC-BY-4.0	Creative Commons Attribution 4.0
CCZ	Clarion–Clipperton Fracture Zone
CNN	Convolutional Neural Networks
COCO	Common Objects in Context
CoMoNoD	Compact-Morphology-based poly-metallic Nodule Delineation
DETR	DEtection TRansformer
DINO	DETR with Improved deNoising anchOr boxes
DL	Deep Learning
GAN	Generative Adversarial Network
IoU	Intersection Over Union
ISA	International Seabed Authority
mAP	mean Average Precision
NWD	Normalized Wasserstein Distance
OFOS	Ocean Floor Observation System
PointRend	Point-based Rendering
ROV	Remotely Operated Vehicle
RPN	Region Proposal Network
SAM	Segment Anything Model
SOLOv2	Segmenting Objects by Locations
YOLO	You Only Look Once

References

1. Abbass, K.; Qasim, M.Z.; Song, H.; Murshed, M.; Mahmood, H.; Younis, I. A review of the global climate change impacts, adaptation, and sustainable mitigation measures. *Environ. Sci. Pollut. Res.* **2022**, *29*, 42539–42559. [CrossRef] [PubMed]
2. Parmesan, C.; Morecroft, M.D.; Trisurat, Y. *Climate Change 2022: Impacts, Adaptation and Vulnerability*; Research Report; UNESCO: Pairs, France, 2022.
3. Agreement, P. Paris agreement. In Proceedings of the Report of the Conference of the Parties to the United Nations Framework Convention on Climate Change (21st Session, 2015: Paris), Paris, France, 30 November–13 December 2015; HeinOnline: Buffalo, NY, USA, 2015; Volume 4, p. 2.
4. Desai, B.H. 14. United Nations Environment Programme (UNEP). *Yearb. Int. Environ. Law* **2020**, *31*, 319–325. [CrossRef]
5. Watari, T.; Nansai, K.; Nakajima, K. Review of critical metal dynamics to 2050 for 48 elements. *Resour. Conserv. Recycl.* **2020**, *155*, 104669. [CrossRef]
6. Hein, J.R.; Koschinsky, A.; Kuhn, T. Deep-ocean polymetallic nodules as a resource for critical materials. *Nat. Rev. Earth Environ.* **2020**, *1*, 158–169. [CrossRef]
7. Sparenberg, O. A historical perspective on deep-sea mining for manganese nodules, 1965–2019. *Extr. Ind. Soc.* **2019**, *6*, 842–854. [CrossRef]
8. Miller, K.A.; Thompson, K.F.; Johnston, P.; Santillo, D. An overview of seabed mining including the current state of development, environmental impacts, and knowledge gaps. *Front. Mar. Sci.* **2018**, *4*, 312755. [CrossRef]
9. Santos, M.; Jorge, P.; Coimbra, J.; Vale, C.; Caetano, M.; Bastos, L.; Iglesias, I.; Guimarães, L.; Reis-Henriques, M.; Teles, L.; et al. The last frontier: Coupling technological developments with scientific challenges to improve hazard assessment of deep-sea mining. *Sci. Total Environ.* **2018**, *627*, 1505–1514. [CrossRef]
10. Minar, M.R.; Naher, J. Recent advances in deep learning: An overview. *arXiv* **2018**, arXiv:1807.08169.
11. Silva, E.; Viegas, D.; Martins, A.; Almeida, J.; Almeida, C.; Neves, B.; Madureira, P.; Wheeler, A.J.; Salavasidis, G.; Phillips, A.; et al. TRIDENT–Technology based impact assessment tool foR sustaInable, transparent Deep sEa miNing exploraTion and exploitation: A project overview. In Proceedings of the OCEANS 2023-Limerick, Limerick, Ireland, 5–8 June 2023; IEEE: New York, NY, USA, 2023; pp. 1–7.
12. Amjoud, A.B.; Amrouch, M. Object detection using deep learning, CNNs and vision transformers: A review. *IEEE Access* **2023**, *11*, 35479–35516. [CrossRef]
13. Song, W.; Dong, L.; Zhao, X.; Xia, J.; Liu, T.; Shi, Y. Review of Nodule Mineral Image Segmentation Algorithms for Deep-Sea Mineral Resource Assessment. *Comput. Mater. Contin.* **2022**, *73*, 1649–1669. [CrossRef]
14. Liu, Y.; Wang, X.; Zhang, Z.; Deng, F. Deep learning in image segmentation for mineral production: A review. *Comput. Geosci.* **2023**, *180*, 105455. [CrossRef]
15. Savaliya, J.; Prabhakaran, K.; Muthuvel, P.; Meenakshi, S.; Varshney, N.; Sankar, P. Underwater Resource Detection Using Image Processing. In Proceedings of the 2023 International Conference on Next Generation Electronics (NEleX), Vellore, India, 14–16 December 2023; pp. 1–5. [CrossRef]
16. Loureiro, G.; Dias, A.; Almeida, J.; Martins, A.; Hong, S.; Silva, E. A Survey of Seafloor Characterization and Mapping Techniques. *Remote Sens.* **2024**, *16*, 1163. [CrossRef]
17. Quintana, J.; Garcia, R.; Neumann, L.; Campos, R.; Weiss, T.; Köser, K.; Mohrmann, J.; Greinert, J. Towards automatic recognition of mining targets using an autonomous robot. In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; IEEE: New York, NY, USA, 2018; pp. 1–7.
18. Sartore, C.; Campos, R.; Quintana, J.; Simetti, E.; Garcia, R.; Casalino, G. Control and perception framework for deep sea mining exploration. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: New York, NY, USA, 2019; pp. 6348–6353.
19. Simetti, E.; Campos, R.; Di Vito, D.; Quintana, J.; Antonelli, G.; Garcia, R.; Turetta, A. Sea mining exploration with an UVMS: Experimental validation of the control and perception framework. *IEEE/ASME Trans. Mechatron.* **2020**, *26*, 1635–1645. [CrossRef]
20. Sun, K.; Wu, Z.; Wang, M.; Shang, J.; Liu, Z.; Zhao, D.; Luo, X. Accurate Identification Method of Small-Size Polymetallic Nodules Based on Seafloor Hyperspectral Data. *J. Mar. Sci. Eng.* **2024**, *12*, 333. [CrossRef]
21. Cui, C.; Ma, P.; Zhang, Q.; Liu, G.; Xie, Y. Grabbing Path Extraction of Deep-Sea Manganese Nodules Based on Improved YOLOv5. *J. Mar. Sci. Eng.* **2024**, *12*, 1433. [CrossRef]
22. Dong, L.; Wang, H.; Song, W.; Xia, J.; Liu, T. Deep sea nodule mineral image segmentation algorithm based on Mask R-CNN. In Proceedings of the ACM Turing Award Celebration Conference-China (ACM TURC 2021), Hefei, China, 30 July–1 August 2021; pp. 278–284.
23. Tomczak, A.; Kogut, T.; Kabała, K.; Abramowski, T.; Ciążela, J.; Giza, A. Automated estimation of offshore polymetallic nodule abundance based on seafloor imagery using deep learning. *Sci. Total Environ.* **2024**, *956*, 177225. [CrossRef]
24. Song, W.; Zheng, N.; Liu, X.; Qiu, L.; Zheng, R. An improved u-net convolutio nal networks for seabed mineral image segmentation. *IEEE Access* **2019**, *7*, 82744–82752. [CrossRef]

25. Park, C.Y.; Chon, H.T.; Kang, J.K. Correction of nodule abundance using image analysis technique on manganese nodule deposits. *Econ. Environ. Geol.* **1996**, *29*, 429–437.
26. Mao, H.; Liu, Y.; Yan, H.; Qian, C.; Xue, J. Image processing of manganese nodules based on background gray value calculation. *Comput. Mater. Contin.* **2020**, *65*, 511–527.
27. Schoening, T.; Steinbrink, B.; Brün, D.; Kuhn, T.; Nattkemper, T.W. Ultra-fast segmentation and quantification of poly-metallic nodule coverage in high-resolution digital images. In Proceedings of the UMI, Rio de Janeiro, Brazil, 21–25 October 2013.
28. Schoening, T.; Jones, D.O.; Greinert, J. Compact-morphology-based poly-metallic nodule delineation. *Sci. Rep.* **2017**, *7*, 13338. [CrossRef]
29. Kuhn, T.; Wegorzewski, A.; Rühlemann, C.; Vink, A. Composition, formation, and occurrence of polymetallic nodules. In *Deep-Sea Mining: Resource Potential, Technical and Environmental Considerations*; Springer: Cham, Switzerland, 2017; pp. 23–63.
30. Purser, A.; Bodur, Y.V.; Ramalo, S.; Stratmann, T.; Schoening, T. *Seafloor Images of Undisturbed and Disturbed Polymetallic Nodule Province Seafloor Collected During RV SONNE Expeditions SO268/1+2*; PANGAEA: Bremerhaven, Germany, 2021. [CrossRef]
31. Haeckel, M.; Linke, P. *RV SONNE Fahrtbericht/Cruise Report SO268-Assessing the Impacts of Nodule Mining on the Deep-Sea Environment: NoduleMonitoring, Manzanillo (Mexico)–Vancouver (Canada), 17.02.–27.05.2019*; GEOMAR Helmholtz-Zentrum für Ozeanforschung Kiel: Kiel, Germany, 2021.
32. Supervisely. Supervisely Computer Vision Platform. 2023 Available online: https://supervisely.com (accessed on 20 July 2023).
33. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. *arXiv* **2023**, arXiv:2304.02643.
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
35. Varghese, R.; Sambath, M. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. In Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 18–19 April 2024; IEEE: New York, NY, USA, 2024; pp. 1–6.
36. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 213–229.
37. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv* **2022**, arXiv:2203.03605.
38. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
39. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *arXiv* **2018**, arXiv:1703.06870. [CrossRef]
40. Wang, X.; Zhang, R.; Kong, T.; Li, L.; Shen, C. SOLOv2: Dynamic and Fast Instance Segmentation. *arXiv* **2020**, arXiv:2003.10152. [CrossRef]
41. Kirillov, A.; Wu, Y.; He, K.; Girshick, R. PointRend: Image Segmentation as Rendering. *arXiv* **2020**, arXiv:1912.08193. [CrossRef]
42. Cheng, B.; Misra, I.; Schwing, A.G.; Kirillov, A.; Girdhar, R. Masked-attention Mask Transformer for Universal Image Segmentation. *arXiv* **2022**, arXiv:2112.01527.
43. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597. [CrossRef]
44. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018.
45. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]
46. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. *arXiv* **2021**, arXiv:2105.15203.
47. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
48. Contributors, M. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. 2020. Available online: https://github.com/open-mmlab/mmsegmentation (accessed on 1 February 2025).
49. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
50. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2019**, arXiv:1711.05101. [CrossRef]
51. Padilla, R.; Netto, S.L.; Da Silva, E.A. A survey on performance metrics for object-detection algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; IEEE: New York, NY, USA, 2020; pp. 237–242.

52. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *arXiv* **2015**, arXiv:1411.4038. [CrossRef]
53. Ogwok, D.; Ehlers, E.M. Jaccard index in ensemble image segmentation: An approach. In Proceedings of the 2022 5th International Conference on Computational Intelligence and Intelligent Systems, Quzhou, China, 4–6 November 2022; pp. 9–14.
54. Shamir, R.R.; Duchin, Y.; Kim, J.; Sapiro, G.; Harel, N. Continuous Dice Coefficient: A Method for Evaluating Probabilistic Segmentations. *arXiv* **2019**, arXiv:1906.11031. [CrossRef]
55. Jocher, G.; Qiu, J.; Chaurasia, A. Ultralytics YOLO. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 2 February 2025).
56. Jie, W.L.; Kalyan, B.; Chitre, M.; Vishnu, H. Polymetallic nodules abundance estimation using sidescan sonar: A quantitative approach using artificial neural network. In Proceedings of the OCEANS 2017-Aberdeen, Aberdeen, UK, 19–22 June 2017; IEEE: New York, NY, USA, 2017; pp. 1–6.
57. Wong, L.J.; Kalyan, B.; Chitre, M.; Vishnu, H. Acoustic assessment of polymetallic nodule abundance using sidescan sonar and altimeter. *IEEE J. Ocean. Eng.* **2020**, *46*, 132–142. [CrossRef]