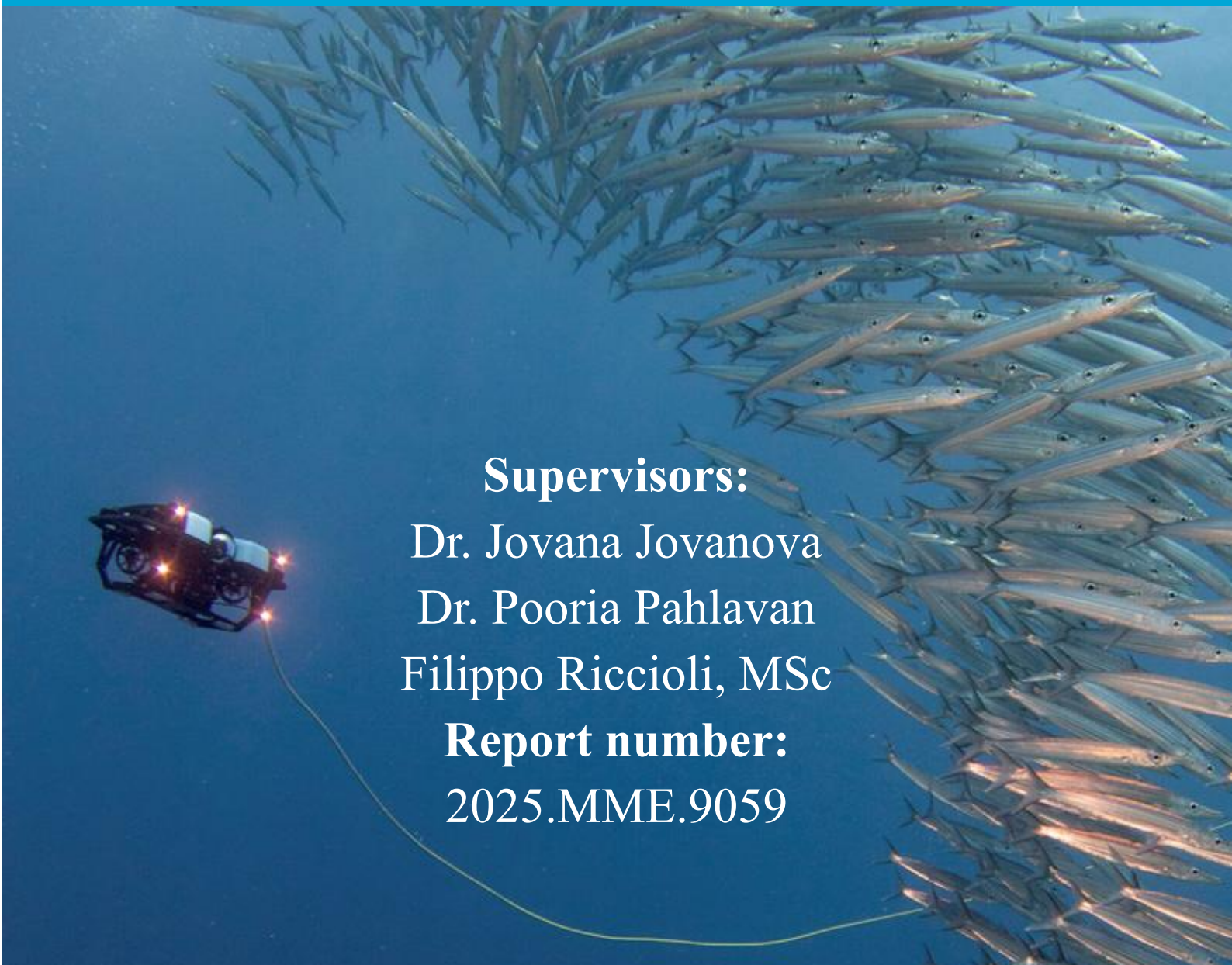
Assisted navigation for underwater robotics

Graduation assignment report

Enrong Xiang

TU Delft



Supervisors:

Dr. Jovana Jovanova

Dr. Pooria Pahlavan

Filippo Riccioli, MSc

Report number:

2025.MME.9059

Assisted navigation for underwater robotics

By

Enrong Xiang

Master Thesis

in partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical Engineering

to be defended publicly on Monday June 2, 2025 at 8:30 AM

Student number: 5936128
MSc track: Multi-Machine Engineering
Report number: 2025.MME.9059

Supervisor:
Dr. Jovana Jovanova
Dr. Pooria Pahlavan
Filippo Riccioli, MSc

Date: May 2025

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

Contents

1. Introduction 1
1.1. Background and motivation 1
1.2. Problem statement 2
1.3. Underwater imaging 3
1.4. Literature review 5
1.4.1. Preliminary knowledge 5
1.4.2. State-of-the-art 23
1.4.3. Underwater localization 29
1.4.4. Knowledge gap 33
1.5. Research questions 34
1.6. Report structure 35
2. Background knowledge 36
2.1. Faster RCNN 36
RCNN 36
Fast RCNN 36
Faster RCNN 36
2.2. Identification false-prevention methods 38
Distance filtering 38
Abnormal object filtering 39
Detected object recalling 40
3. Methodology 43
3.1. Image enhancement 43
3.2. Underwater identification 46
3.2.1. Object identification 46
3.2.2. Object labelling 53
3.3. Underwater localization 54
3.3.1. Method 1 54
3.3.2. Method 2 60
4. Experiments 67
4.1. Experiment setup 67
4.2. Experiment procedure 68
5. Experimental results 68
6. Conclusions and recommendations 73
6.1. Conclusions 73
6.2. Recommendations 74
Appendix 76
A. Scientific research paper 76
B. Grabber mechanism 85
TPU grabber 86
PLA grabber 89
C. GUI 94
Underwater identification & Localization 94

Load video 94
Visualization 95
D. Image enhancement results 98
E. Running environmental requirements 101
F. Training dataset for deep learning 103
G. Alternative method for localization 106
H. Alternative models for identification 109
Two-stage algorithm 109
One-stage algorithm 110
I. Alternative depth estimation algorithm 113
MiDas 113
Depth Anything v2 114
J. Validation 116
Training iteration exploration 116
K. Underwater operational scenario videos 122
Bibliography 123

Abstract

Underwater detection has garnered increasing attention in recent years due to its broad and impactful applications in marine ecological research, underwater structural inspection, archaeological exploration, and deep-sea resource extraction. However, despite the proliferation of research in this domain, a comprehensive methodology that addresses both object identification and localization in underwater scenarios remains absent. Existing studies tend to treat these two tasks separately, often omitting the practical implementation details necessary for real-world deployment. This fragmented approach limits the effectiveness and adaptability of underwater detection systems, particularly in dynamic or unpredictable marine conditions.

To bridge this gap, this report presents a detailed exploration of a camera-based framework for simultaneous object identification and localization in underwater environments. The proposed system leverages a Region-based Convolutional Neural Network (RCNN) for object identification, offering a favorable trade-off between precision and computational efficiency. RCNN's architecture enables it to effectively handle the complex visual features typically present in underwater imagery. For the localization task, two complementary strategies are employed: the Metric3D depth estimation algorithm, which utilizes learned monocular cues to infer depth maps with high accuracy, and a geometry-based method rooted in camera imaging principles, which estimates object distance based on intrinsic and extrinsic camera parameters. The former localization method (Metric 3D) is more computationally expensive, but it provides more robust results and easier applications.

Experimental evaluations demonstrate that the proposed integrated approach achieves robust performance in various underwater conditions. The RCNN consistently delivers accurate object classifications, while the localization strategies offer flexibility and reliability depending on the computational and environmental constraints.

Overall, this research contributes a novel and practical solution for real-time underwater object detection by unifying identification and localization. The proposed system enables safer navigation, more precise manipulation, and greater situational awareness. By addressing the methodological gaps in existing literature and emphasizing real-world applicability, this work contributes to the research of intelligent underwater operation and automation.

1. Introduction

1.1. Background and motivation

The underwater environment poses significant challenges for robotic operations due to limited visibility, unpredictable terrain, and the absence of reliable reference systems. Accurate localization and identification of underwater objects are essential to support autonomous or semi-autonomous underwater missions. These capabilities can enhance navigation, enable targeted operations, and provide valuable references for subsequent tasks.

However, underwater localization is inherently difficult. GPS signals cannot penetrate water, and installing underwater positioning systems is much more complex compared to terrestrial environments. Additionally, the vast diversity of underwater objects makes manual classification time-consuming and inefficient. To address these challenges, various techniques have been proposed for underwater localization and object identification. Nevertheless, most existing approaches treat these two tasks separately, with few systems implementing both simultaneously in a cohesive manner. A summary of commonly used identification and localization techniques is presented in Table 1.

Table 1 Some underwater identification and localization methods [1, 2, 3, 4, 5, 6].

No.	Title	Contents
1	Sonar image registration for localization of an underwater vehicle	This study proposes a system that enhances the localization capabilities of an Autonomous Underwater Vehicle (AUV) by utilizing side-scan sonar data.
2	Towards mapping of underwater structures by a team of autonomous underwater vehicles	This work presents a method for generating high-resolution and accurate positional representations of underwater structures, facilitating detailed mapping and analysis.
3	Underwater localization and 3D mapping of submerged structures with a single-beam scanning sonar	This paper introduces a multi-stage pipeline for localization and 3D mapping in cluttered, shallow-water environments, relying solely on a depth sensor and a single-beam scanning sonar.
4	Underwater Object Detection in the Era of Artificial Intelligence: Current, Challenge, and Future	This study offers a comprehensive analysis of artificial intelligence-based approaches for underwater object identification, including both traditional machine learning techniques and modern deep learning methodologies.
5	Underwater object detection: architectures and algorithms – a comprehensive review	This paper proffered an understandable and profound review of highly renowned deep learning algorithms for underwater object identification.

6	Real-Time Underwater Maritime Object Detection in Side-Scan Sonar Images Based on Transformer-YOLOv5	To address the limitations of manual detection in side-scan sonar (SSS) imagery, this paper proposes a real-time Automatic Target Recognition (ATR) method that enhances detection efficiency and accuracy.

A variety of sensors are available for underwater object localization and identification, including imaging sonar, magnetic sensors, and electromagnetic sensors. Among these, imaging sonar is one of the most widely adopted due to its extensive detection range. For instance, the Ping360 Scanning Imaging Sonar integrated with the BlueROV2 can detect objects at distances of up to 50 meters [7]. However, while effective for long-range detection, imaging sonar typically provides limited information regarding object appearance and classification.

In contrast, underwater cameras are increasingly favored for their ability to capture rich visual details, making them well-suited for both object identification and localization tasks. The integration of underwater image enhancement techniques further mitigates issues such as low contrast, color distortion, and turbidity, thereby improving the quality and usability of captured imagery. Owing to this balance between spatial accuracy and semantic richness, cameras are often the preferred choice for vision-based underwater inspection systems.

1.2. Problem statement

Object identification and localization are fundamental components of underwater robotic operations, as they enable informed decision-making, support autonomous navigation, and mitigate potential operational risks. Despite their critical importance, these two functionalities are seldom integrated into a unified framework. Existing literature on underwater object identification has predominantly focused on the application of machine learning and deep learning techniques for recognizing and classifying objects in underwater imagery. However, these studies frequently overlook the spatial localization of the identified objects.

Conversely, research on underwater localization typically relies on technologies such as imaging sonar, electromagnetic sensors, or acoustic positioning systems, including Short Baseline (SBL) and Ultra-Short Baseline (USBL) systems. These approaches tend to prioritize positional accuracy without incorporating object identification capabilities. This highlights a significant gap in current underwater robotic research.

A literature review using search terms such as “simultaneous identification and localization” reveals that the majority of results are centered around Simultaneous Localization and Mapping (SLAM), which primarily addresses environmental mapping rather than explicit object recognition. As shown in Figure 1, there is a clear lack of research that combines object identification and localization, highlighting the need for more integrated approaches in this area.

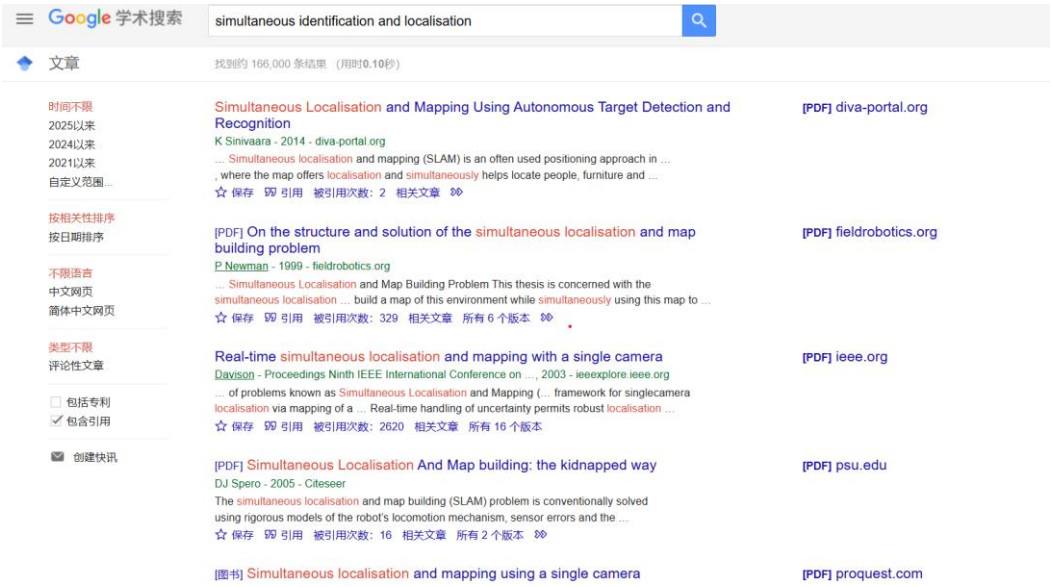


Figure 1. the search results of 'Simultaneous identification and localization' in Google scholar.

Moreover, the limitations of certain underwater sensors pose challenges to achieving precise operations. For example, while imaging sonar provides broad coverage, it often lacks high accuracy at close range. Similarly, magnetic and electromagnetic sensors are constrained by their very limited detection ranges. These shortcomings hinder the ability to accurately determine both the category and position of underwater objects, thereby reducing operational efficiency. In contrast, underwater cameras offer a favorable balance between detection range and resolution, capturing rich visual data suitable for both identification and localization tasks. As a result, the development of a vision-based system capable of simultaneously performing object identification and localization is not only feasible but also highly advantageous for practical underwater applications.

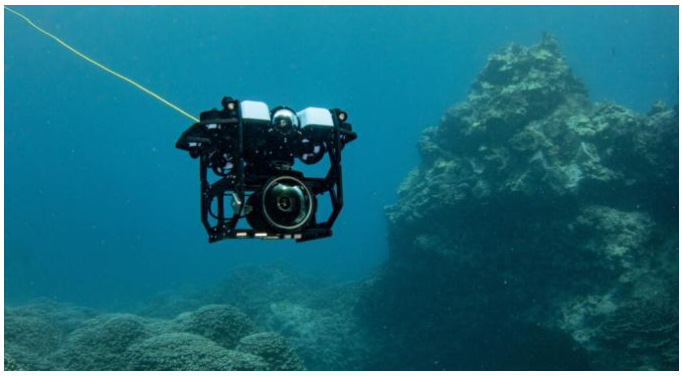
1.3. Underwater imaging

Imaging refers to the visual representation or reproduction of an object's form, typically captured using a camera. While aerial imaging is extensively applied in fields such as aerial mapping [8], land-based target detection and tracking, and vegetation supervision [9], underwater imaging introduces a distinct set of challenges that require specialized solutions. One major limitation is the waterproofing capability of imaging equipment. Most conventional electronic cameras are not inherently water-resistant and may malfunction or sustain damage in submerged conditions. Even devices designed for underwater use have operational depth limits; for instance, the GoPro HERO12 is only rated for depths of up to 10 meters. Another significant challenge is poor visibility. Underwater environments frequently impair image quality due to factors such as color deviation, low contrast and non-uniform illumination [10]. These conditions necessitate the use of advanced imaging devices or supplementary enhancement techniques to improve image clarity and usability. An example of a degraded underwater image is

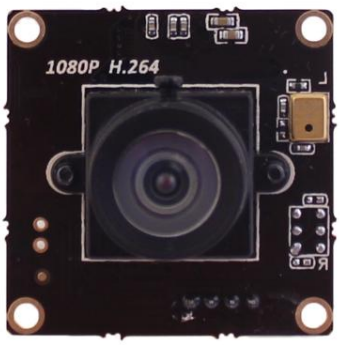
shown in Figure 2.

Figure 2. Blurry and Bluish Effects from Haze and Color cast in Underwater Images [11].

To address these challenges, marine vision systems are increasingly using cameras that are specially designed for underwater use. For example, the BlueROV2, developed by Blue Robotics, is equipped with a Low-Light HD USB Camera, which has excellent low-light performance, good color handling, and onboard video compression. The underwater camera is indicated in Figure 3.



(a) Camera equipped with the BlueROV2



(b) The Low-Light HD USB Camera

Figure 3. The underwater camera [12].

Another critical parameter in underwater imaging is the field of view (FOV), which influences the spatial extent captured by the camera. The FOV can be described by the following equation:

$$\frac{p}{f} = \frac{l}{d} \tag{1}$$

Where p is the sensor dimension, f is the focal length of the camera. l is the field dimension, d is the distance to field. A schema representation is shown in Figure 4. In equation (1), if the field dimension l and distance to field d are predefined, the required focal length and sensor dimension can be obtained. The illustration of the field of view is indicated in Figure 4.

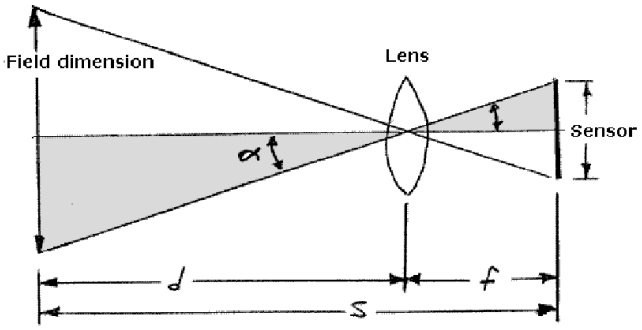


Figure 4. A schema representation of field of view [13]. The 'sensor' in the picture refers to sensor dimension p

In addition to specially designed underwater cameras, the image enhancement technique can also improve underwater image quality. This technique will be further discussed in Section 3.1. Image Enhancement.

1.4. Literature review

In Chapter 1.4, the literature review for underwater identification and localization is presented below.

1.4.1. Preliminary knowledge

Before the introduction of state-of-the-art, some core concepts are explained in advance. The explanations will facilitate the understanding of the state-of-the-art.

Underwater identification

Image processing

ROI (Regions of interest)

Region of interest (ROI) plays an important role in image analysis, which means the regions of interest are isolated from the background. There are two advantages, Firstly, the computational complexity can be reduced, because in subsequent steps, only the regions that are extracted will be processed. Secondly, precision can be increased,

because the errors are controlled [14]. There are various methods that can realize ROI. For example, Huang, C. et al. [15] proposes a new method for regions of interest extraction from color image based on visual saliency in HSV color space. Zhang, J. et al. [16] proposes an approach for content-based image retrieval combining both color and texture features using three ROIs. Wang, Z. et al. [17] developed an auxiliary Gaussian weighting (AGW) scheme, which is incorporated into the ROI based image retrieval system. One example of ROI detection is indicated in Figure 5.

Figure 5. ROI detection [18]. The bounding box on the marine animal has a different color compared to others

Underwater image enhancement

Underwater image enhancement aims to improve the quality of image, including eliminating noise and color distortion, enhancing features of interest, weakening irrelevant background features. It has been widely applied in computer vision tasks, remote sensing and surveillance. There are several approaches to realize image enhancement. It can generally be divided into Non-physical Model-based methods, Physical Model-based methods, where it considers the enhancement as an inverse process of underwater imaging, it mainly employs the prior knowledge and optical properties of underwater imaging. Finally, there comes the data-driven methods, which are mainly inspired by deep-learning [19]. In this literature, the image preprocessing is mostly realized through image enhancement. The effectiveness of image enhancement based on GAN is indicated in Figure 6.

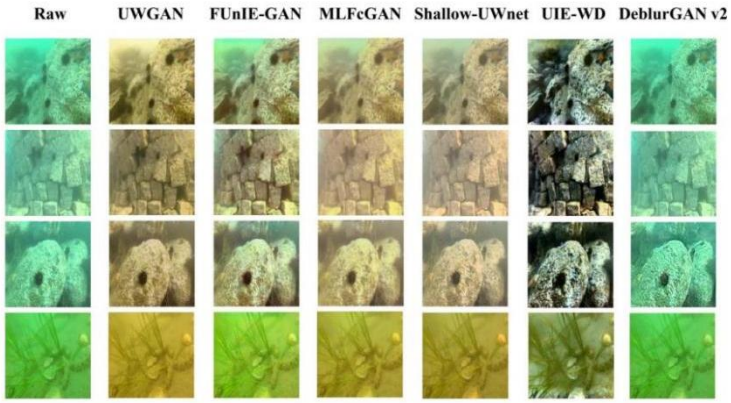


Figure 6. The enhancement results of several GAN-based underwater image algorithms, from left to

right: Raw images, UWGAN [20], FUnIE-GAN [21], MLFcGAN [22], Shallow-UWnet [23], UIE-WD [24], DEblurGANv2 [25].

Generative Adversarial Network

The goal of a generative model is to study a collection of training examples and learn the probability distribution that generated them. As a common image enhancement algorithm, Generative Adversarial Networks (GANs) are then able to generate more examples from the estimated probability distribution. GANs are among the most successful generative models based on deep learning [26].

Image segmentation

Image segmentation is a key task in computer vision and image processing with important applications such as scene understanding, medical image analysis, robotic perception, video surveillance, augmented reality, and image compression, among others [27]. Image segmentation divides images into regions with different features and extracts the regions of interest (ROIs). These regions, according to human visual perception, are meaningful and non-overlapping [28]. Lots of research on image segmentation has been carried out, which facilitates its development. The existing image segmentation methods are indicated in Figure 7.

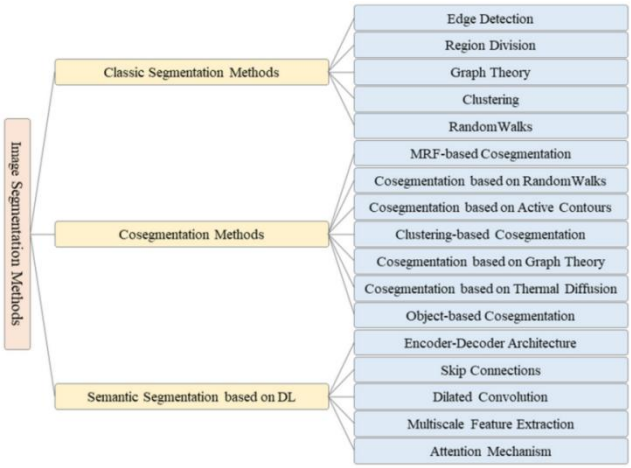


Figure 7. The existing image segmentation methods [28].

Iterative fuzzy segmentation

The algorithm is designed to extract precise shadow contours from noisy images. It begins by applying threshold-based segmentation to produce an initial segmented image. In the subsequent iterative step, a membership function is used to refine the shadow contour pixels. This function evaluates pixel properties by combining two fuzzy functions: one that assesses pixel brightness and another that considers connectivity based on the pixel's immediate neighborhood.

Underwater Image Saliency Detection

Underwater saliency detection is intended to equip computers with the ability to efficiently distinguish and perceive salient objects in images or videos to comprehend underwater scenes. Underwater image saliency detection aims to identify the most

noticeable object(s) in an image, which can be applied to summarize images and present the most eye-catching things to observers. SOD is different from object detection and semantic segmentation, which seeks to locate all the objects in the image. SOD also produces dense, pixel-level labels but with only two classes, the prominent and non-prominent parts [29]. Figure 8 shows examples of an image's SOD, object detection, and semantic segmentation results for their differences.

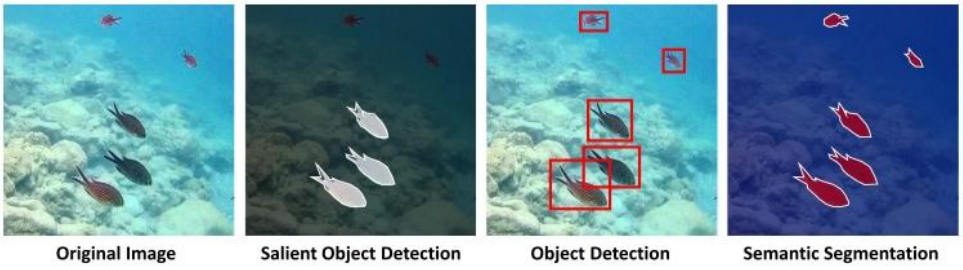


Figure 8. Results of SOD, object detection, and semantic segmentation of an image [29]. SOD produces labels with only two classes, the prominent and non-prominent parts

There is also some relevant research on this topic. Jian, M. et al. [30] proposes a novel framework for underwater image saliency detection by exploiting Quaternionic Distance Based Weber Descriptor (QDWD), pattern distinctness, and local contrast. The proposed algorithm incorporates quaternion number system and principal components analysis (PCA) simultaneously, so as to achieve superior performance.

Classification

Transfer learning

Machine learning typically requires large quantities of training data, sometime the training data can be difficult to obtain. Therefore, there is a need to create high-performance learners trained with more easily obtained data from different domains. This methodology is referred to as transfer learning.

Figure 9 shows the difference between the learning processes of traditional and transfer learning techniques. As we can see, traditional machine learning techniques try to learn each task from beginning, while transfer learning techniques try to transfer the knowledge from some previous tasks to a target task that requires high-quality training data [31].

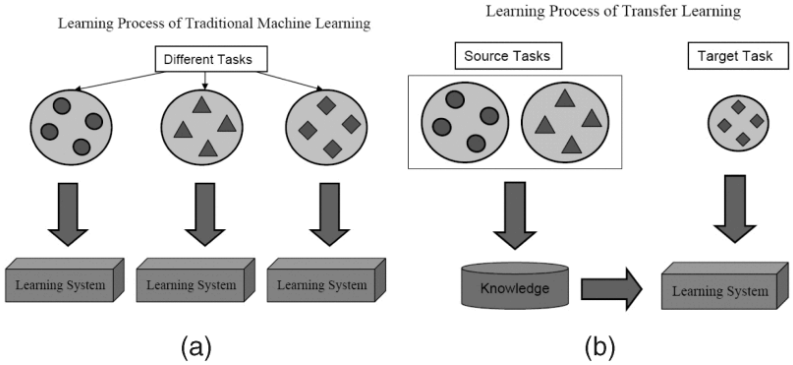


Figure 9. Different learning processes between (a) traditional machine learning and (b) transfer learning [31].

Machine learning methods for underwater object detection

For machine learning methods for underwater object detection, they generally consist of two-stage learning. In the first stage, hand-crafted features are extracted, which can include simple visual features (e.g., color and shape) or complex hand-crafted features (e.g., HOG, SIFT and SURF). In the second stage, these features are forwarded to traditional classifiers, such as SVM and decision trees, to carry out various classification tasks in underwater scenes. This method can be further classified into sonar based method and camera based method. The sonar based methods have been widely applied in underwater exploration, as they can provide relatively reliable scene data regardless of visibility. In contrast to sonars, camera based method can capture a large number of RGB images with high spatial and temporal resolutions. Advantages and disadvantages of sonar and camera based method are shown in Figure 10.

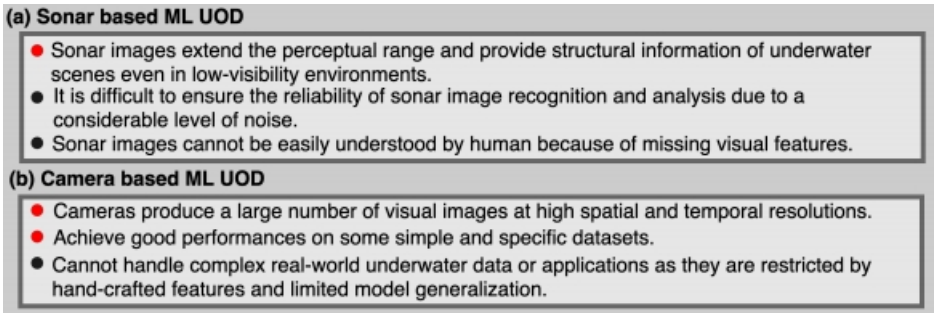


Figure 10. Comparison between sonar-based method and camera based method [4].

Support vector machine

Support Vector Machine (SVM) is a machine learning method that has become exceedingly popular. It has been widely applied in text and hypertext categorization, classification of images and neuroimaging analysis in recent years. Effective use of SVM in neurosciences does not require an in-depth understanding of its mathematical foundation, but it does demand a clear conceptual understanding and conscientiousness with respect to application. The process of training an SVM decision function amounts to identifying a reproducible hyperplane that maximizes the distance (i.e., the "margin") between the support vectors of both class labels. The schema of the classification is indicated in Figure 11.

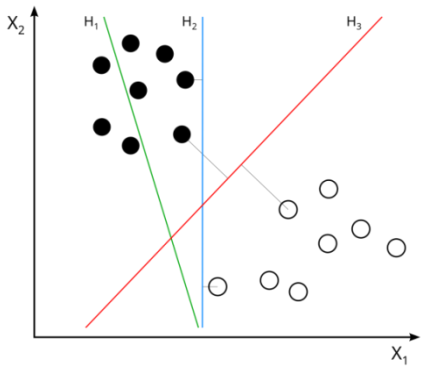


Figure 11. The classification for SVM [32]. X1 represents feature set 1, X2 represents feature set 2. H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximal margin. Therefore, H3 is the good reproducible hyperplane

Random Forest

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that works by creating a multitude of decision trees during training. To carry out the classification, the random forest classifier is developed, it consists of a combination of tree classifiers where each classifier is generated using a random vector sampled independently from the input vector, and each tree casts a unit vote for the most popular class to classify an input vector [33]. Specific process of random forest is indicated in Figure 12.

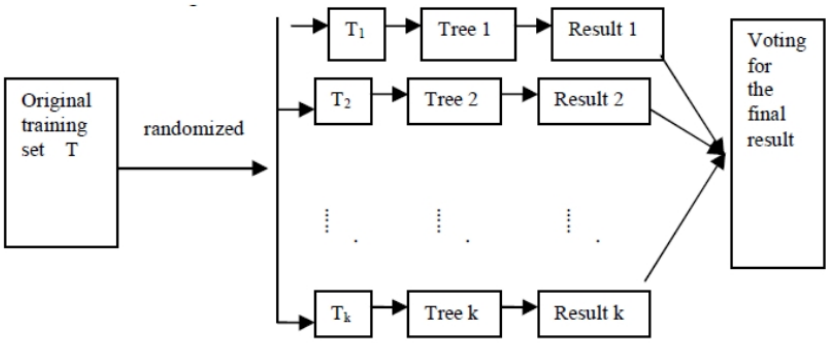


Figure 12. Random forest schematic [34]. The result of each tree represents a vote for a specific class, The more votes one class gets, the more likely the input belongs to that class. The input is more likely to belong to a specific class if it obtains more notes than the other classes

K-Nearest Neighbors

The k-Nearest-Neighbours (kNN) is a non-parametric classification method, which is simple but effective in many cases. The key idea of a standard kNN method is to predict the label of a test data point by the majority rule, that is, the label of the test data point is predicted with the major class of its k most similar training data points in the feature space [35]. The main challenge of KNN algorithm is the selection of the k value due to the fact that setting all test data with the same k value has been proven to be impractical in real applications. Therefore, Guo, G. et al. [36] proposes a novel kNN

type method constructing a kNN model for the data, which replaced the data to serve as the basis of classification. The value of k was automatically determined, based on different data, and is optimal in terms of classification accuracy. The schema of KNN is indicated in Figure 13.

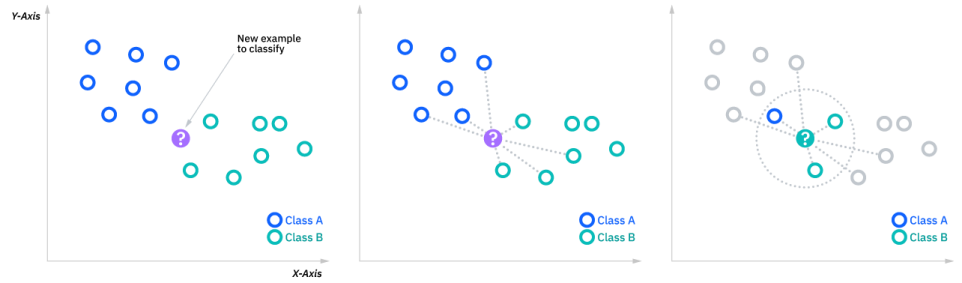
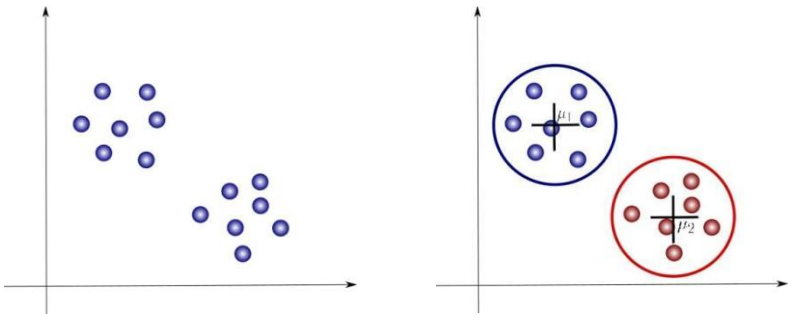


Figure 13. Schema of KNN [37]. Similar training data points are next to the new example to be classified. According to the figure on the right, the K is 3, and the result should be Class B, because more points belong to Class B compared to Class A

Gaussian mixed model

A Gaussian mixture model (GMM) is a machine learning method used to determine the probability each data point belongs to a given cluster. The model is a soft clustering method used in unsupervised learning [38]. In unsupervised learning problems, clustering is an important carrier where clusters of points in the data set that share common characteristics can be found. Finally, the classification can be realized based on the clusterings. The principle of GMM is shown in Figure 14.



(a) Unclassified points (b) Classified points

Figure 14. The schema of GMM [39]. The points are classified based on the characteristics of clustering

Deep learning methods

Deep learning is a subset of machine learning that uses multilayered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain. Neural networks, or artificial neural networks, attempt to mimic the human brain through a combination of data inputs, weights and bias—all acting as silicon neurons. These elements work together to accurately recognize, classify and describe objects within the data, it has been widely applied in remote sensing,

agriculture production, medical science, robotics, healthcare, object identification, human action recognition, speech recognition and so on [40], making it an extremely popular technology nowadays. A gathering of deep learning techniques is shown in Figure 15.

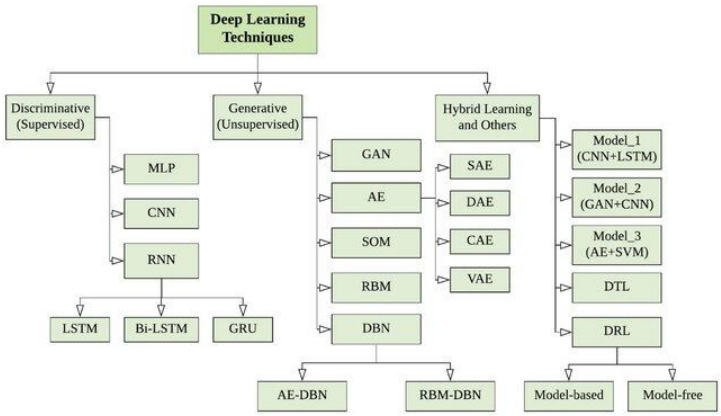


Figure 15. A gathering of deep learning techniques [41]

The deep learning method for underwater object detection belongs to CNN. It can be divided into two categories, one-stage and two-stage. Two-stage algorithms include RCNN series, which are RCNN, Fast RCNN and Faster RCNN. It divides the detection task into two steps: candidate region extraction and classification, which are characterized by high detection accuracy. One-stage algorithms are characterized by faster speed, which are suitable for real-time detection scenarios, it includes Yolo series, SSD, RetinaNet etc. In the subsequent parts, the most representative algorithms, CNN, RCNN series and Yolo series are introduced.

Convolutional neural network

A convolutional neural network (CNN) is one of the most significant networks in the deep learning field. CNN has made impressive achievements in many areas, including but not limited to computer vision and natural language processing. CNN can realize feature extraction. Different from the traditional feature extraction methods, CNN does not need to extract features manually. Therefore, feature extraction is easier. CNN is also a classifier that can realize precise object classification, which makes it a popular technique for classification.

Apart from more efficient feature extraction and precise classification, there are also other advantages. For example, compared with traditional fully connected (FC) networks, each neuron is connected to much fewer neurons, which effectively reduces the parameters and speeds up the convergence. The connection for FC networks and CNN networks is indicated in Figure 16.

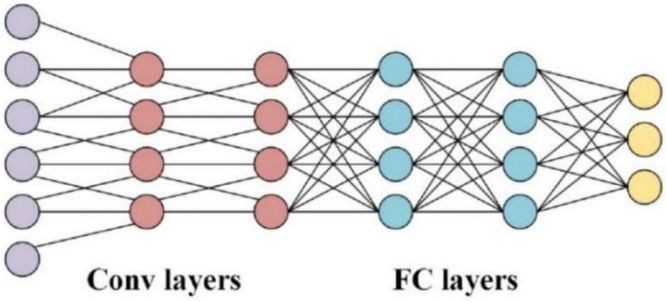


Figure 16. Diagram of CNN layers and FC layers [42]. Compared with traditional fully connected (FC) networks, each neuron is connected to much fewer neurons in CNN

In view of the lower computational burden, CNN becomes one of the most representative algorithms in the deep learning field.

RCNN series

RCNN

Unlike CNN, which produces a single class label for the entire image, RCNN produces multiple class labels and bounding boxes within the image, it proffers a bunch of boxes in the underwater picture and examines if any one of the boxes has an underwater object. The RCNN model does not work in a huge number of regions. It performs a selective survey to draw out these boxes from an underwater picture and these extracted boxes are known as regions [43].

Fast RCNN

To reduce the computational time of RCNN, a CNN model is made to run just once per picture instead of making it run 2000 times per picture. All the ROIs (portions in an image that contain some underwater objects) are then obtained. The Fast RCNN is more time and cost efficient than RCNN.

Faster RCNN

The Faster RCNN framework has two subnetworks namely, Fast RCNN and the Region Proposal Network (RPN). The Faster RCNN network employs Region Proposal Network (RPN). RPN uses feature maps of pictures as input and produces a set of proposals of an object, each having score of objectness as an output. It is found that the Faster RCNN has a higher efficiency compared to Fast RCNN.

YOLO series

The YOLO model uses the complete image in one instance and performs the prediction of the boundary box coordinates and for these bounding boxes, it also predicts the class probabilities. The major benefit of employing the YOLO framework is its extremely high speed.

Yolo-V1

The core principle proposed by YOLO-v1 was the imposing of a grid cell with dimensions of s×s onto the image. In the case of the center of the object of interest falling into one of the grid cells, that particular grid cell would be responsible for the detection of that object. This permitted other cells to disregard that object in the case of multiple appearances [44]. Thus, this algorithm is unbelievably fast.

Yolo-V2

The version of YOLO-V2 focuses mainly on the improvisation of localization and recall while also maintaining precision in object classification, However, it is still bad with small objects [43].

Yolo-V3

Yolo-v3 is comparatively faster than the previous versions. Similar to YOLO-v2 or YOLO9000, this version also performs the prediction of bounding boxes employing dimension clusters as anchor boxes.

Yolo-V4

YOLO-v4 was essentially the distillation of a large suite of object detection techniques, tested and enhanced for providing a real-time, lightweight object detector. The main aim of YOLO-v4 is the fast speed for performing operations of a neural network and optimization for parallel computations.

Sub-technologies in deep learning

To further deal with some challenges in deep learning, relevant techniques are proposed to deal with the challenges in deep learning, which are listed below.

Feature fusion

Small object detection is difficult in the underwater detection scenario. Therefore, recently, lots of efforts have been devoted to small object detection. The feature fusion is one representative technique [45]. The first canonical and the fundamental solution for multi-scale representation is the feature pyramid network (FPN). As illustrated in Figure 17, high-level feature maps are upsampled and added to lower-level ones in order to generate an information-rich representation that is significant for detecting and classifying small objects.

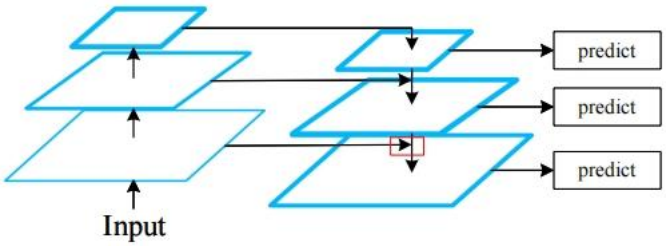


Figure 17. Structure of FPN [46]. High-level feature maps are upsampled and added to lower-level ones in order to facilitate the small object detection and classification

Traditionally, YOLO V4 algorithm and attentional feature fusion has been proposed to address this problem, to produce a harmonious balance between accuracy and speediness for target detection in marine environments. To better enhance the feature fusion, Modified Attentional Feature Fusion (AFFM) module is proposed to better fuse semantic and scale-inconsistent features and to improve accuracy [47].

Contextual information

Leveraging the relationship between an object and its coexisting environment in the real world, contextual information is another novel method to improve small object detection accuracy. The medium and large objects could provide sufficient ROI features in generic detectors. However, it is much necessary to extract more additional

contextual information for small objects as the supplement of original ROI features because the ROI features extracted from the small objects are so few. Some detection methods based on contextual information were proposed to leverage the relationship between small objects and other objects or the background so that the object can be distinguished from the background [48]. Three kinds of contextual information-based small object detection network are presented in Figure 18.

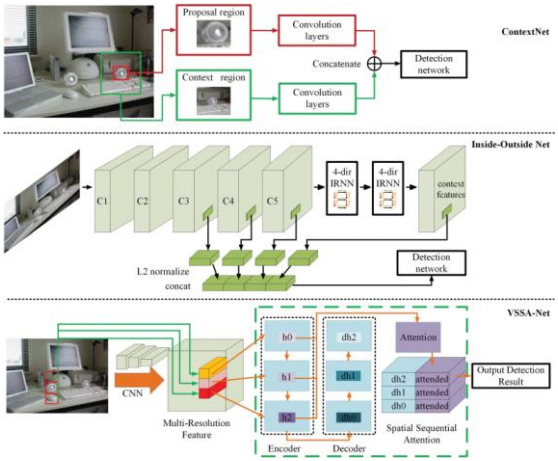


Figure 18. Three kinds of contextual information-based small object detection network, ContextNet, Inside − Outside Net and VSSA-net [48].

Data augmentation

The underwater detection task is usually negatively influenced by lack of clean data. Data augmentation, whose main goal is to increase the volume, quality and diversity of training data, can be used to solve the problem. Input space (or sample space) data augmentation techniques are a class of approaches that directly modify the input image (or parts of it) for the purpose of creating variability to improve generalization. The main advantage of sample space data augmentation techniques is that they are more intuitive and can be designed to specifically generate desired augmentations for a particular task using simple transformation operations. These transformations can operate on the entire image as a whole or can be applied in a patch-wise manner, where the input image is first subdivided into smaller regions before applying transformation on designated or random patches [49]. A scheme for data augmentation is indicated in Figure 19.

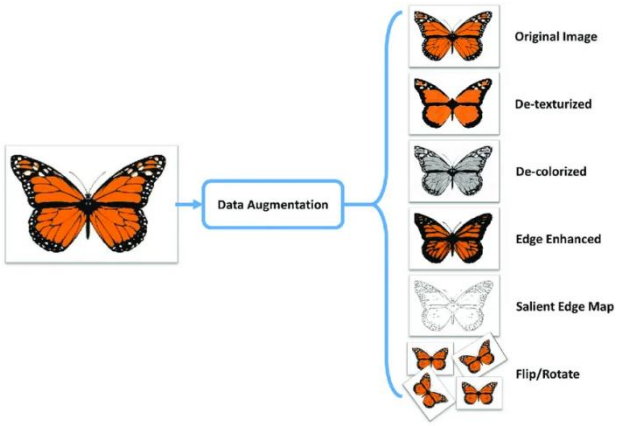


Figure 19. Data augmentation [50]. It can be found that various features are changed from left side to right side of picture

Domain Transformation

Domain transformation refers to the process of performing coordinate transformation on an image, where the transformation can be either global or local, which is an opportunistic strategy for archiving good generalization in underwater object detection by transforming images from distinct domains to middle representations or transforming the trained detector from the source domain to a target domain, as illustrated in Figure 20.

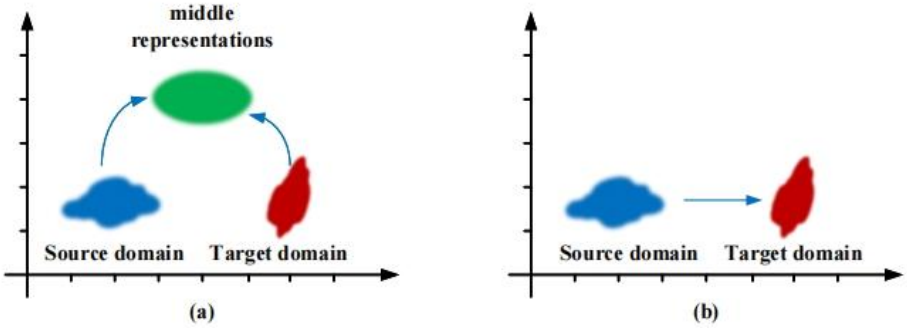


Figure 20. Two kinds of domain transformation for generalization in underwater object detection, where (a) transforms distinct domains to middle representations and (b) transform source domain to target domain [46].

In some research, for example, by adopting the GAN-based restoration approach of [51], Chen, X. et al. successfully bypassed the problem of domain shift. Other GAN-based methods also managed to solve the problem of domain shift, indicating that this is a feasible method.

Underwater localization

Visual-inertial localization methods

Kalman Filter based methods

This method mainly includes two methods, which are Multi-State Constraint Kalman

Filter (MSCKF) and Robust visual inertial odometry (ROVIO).

MSCKF is to track the 3D pose of the IMU-affixed frame {I} with respect to a global frame of reference {G}. In order to simplify the treatment of the effects of the earth's rotation on the IMU measurements, the global frame is chosen as an Earth-Centered, Earth-Fixed (ECEF). The IMU measurements are processed immediately as they become available, for propagating the EKF state and covariance, On the other hand, each time an image is recorded, the current camera pose estimate is appended to the state vector. Finally, when the feature measurements of a given image become available, an EKF update is performed [52]. The general workflow of the MSCKF is indicated in Figure 21.

ROVIO is one of the state-of-the-art monocular visual inertial odometry algorithms. It uses an Iterative Extended Kalman Filter (IEKF) to align visual features and update the vehicle state simultaneously by including the feature locations in the state vector of the IEKF. This algorithm is single-core intensive, which allows the other cores to be used for other algorithms, such as object detection and path optimization [53].

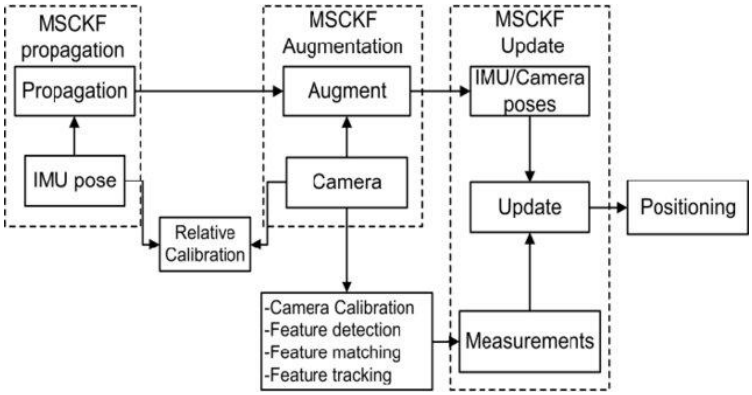


Figure 21. The general workflow of the MSCKF [54]. When IMU pose and camera pose are available for a given image, an EKF update is performed to update the positioning

Cross-correlation

Cross correlation is a standard method of estimating the degree to which two series are correlated. Consider two series x(i) and y(i) where i=0,1,2...N-1. The cross-correlation r at delay d is defined as [55]:

$$r = \frac{\sum_i[(x(i)-mx)*(y(i-d)-my)]}{\sqrt{\sum_i(x(i)-mx)^2}\sqrt{\sum_i(y(i-d)-my)^2}} \quad (2)$$

Where mx and my are the means of the corresponding series. The advantage of the cross-correlation is the best robustness to symmetric additive noise [56]. Therefore, it is unsurprising that the normalized cross-correlation has been used extensively for many signal processing applications, for example, underwater localization.

Triangulation

In trigonometry and geometry, triangulation is the process of determining the location of a point by forming triangles to the point from known points. In computer stereo vision system, the triangulation is the process of determining the positions of 3D

points from 2D mage point correspondences [57], which is based on its angles of arrival from two or more observation points. It has been widely applied in surveying, navigation, metrology, astrometry, binocular vision, model rocketry and the military.

To simplistically describe the principle, the system is composed of an emitter, a receiver and an inspected surface. The emitter emits signals, which is then reflected by the inspected surface. Finally, the receiver receives the signal and further processes the location. The schema representation is indicated in Figure 22.

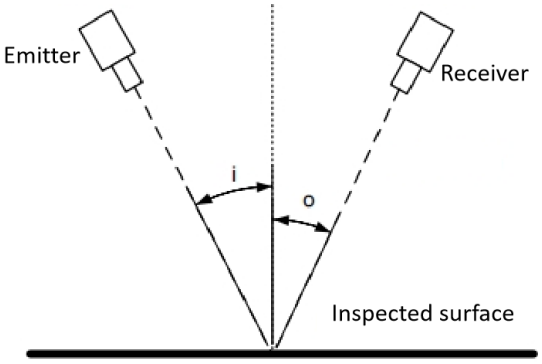


Figure 22. The principle of triangulation [58].

To describe the principle of triangulation in more detail. The illustration of triangulation realized by camera is indicated in Figure 23.

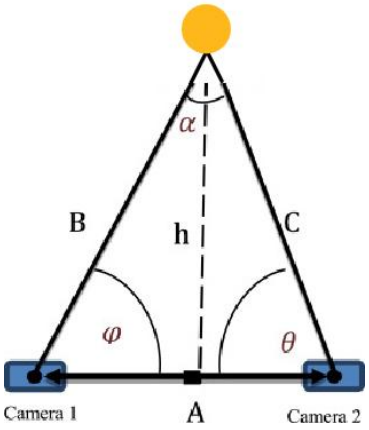


Figure 23. The principle of triangulation realized by depth camera [59].

In this scenario, the cameras are monocular cameras, the system can calculate the angle between the camera-object line and the stereo baseline defined by the two cameras, which are φ and θ. The angles can be expressed as:

$$sin\varphi = \frac{h}{B} \tag{3}$$

$$sin\theta = \frac{h}{C} \tag{4}$$

$$\alpha = \pi - \varphi - \theta \tag{5}$$

Where φ and θ are angles between the camera-object line and the stereo baseline, α is the angle between the camera-object lines, B and C are the distance from the camera 1 and camera 2 to the object respectively. h is the distance from the depth camera to the object. Therefore, the depth h can be written as:

$$\mathrm{h} = \mathrm{B}\sin\varphi = \mathrm{C}sin\theta \tag{6}$$

According to the law of sines, the equation below can be obtained.

$$\frac{A}{\sin\alpha} = \frac{B}{\sin\theta} \tag{7}$$

$$\mathrm{B} = \frac{A\sin\theta}{\sin\alpha} \tag{8}$$

Where A is the distance between the monocular cameras. Finally, the distance from the depth camera to the object can be obtained below:

$$\mathrm{h} = \frac{\mathrm{A}\sin\theta \cdot sin\varphi}{\sin\alpha} \tag{9}$$

Filtering algorithms

Median filter

The median filtering algorithm has good noise-reducing effects. It ranks all the pixels in the kernel in terms of brightness by sliding a window over the image [60] and then changes the value of the pixel in question to be the same as the median, or center value, of this ranking. The net result is that the median filter does an excellent job of removing single-pixel noise artifacts, while causing only a slight reduction in the sharpness of the image [61].

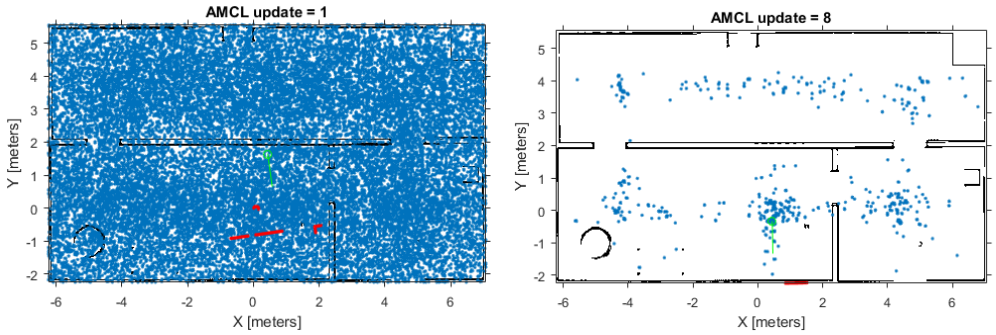
As mentioned before, in the median filter, the questioned pixel is replaced by its neighborhood median. It is an order statistics filtering process. The equation is written as follows [62]:

$$\mathrm{f}(\mathrm{x},\mathrm{y}) = \mathrm{median}[\mathrm{g}(\mathrm{s},\mathrm{t})],\ (\mathrm{s},\mathrm{t}) \in \mathrm{S}(\mathrm{x},\mathrm{y}) \tag{10}$$

Where f(x,y), the filtered image depends on the ordering of the pixel values of the image g(s, t) and the noisy image in the window S(x, y). $\mathrm{S}(\mathrm{x},\mathrm{y}) = \mathrm{n}(\mathrm{x}+\mathrm{s},\mathrm{y}+\mathrm{t}), \mathrm{s} = \mathrm{a}, \mathrm{s} = \mathrm{b}$

Monte Carlo Algorithm

Monte Carlo Algorithm (MCL), also known as particle filter localization, generally requires a metrical map of the environment to calculate a robot's position from the posterior probability density of a set of weighted samples, where image-based localization matches a robot's current view of the environment with reference views. The mobile robot can localise and orient itself as it moves through an environment [63]. The Monte Carlo Algorithm utilizes Bayesian filtering (also known as Markov localization in robotics) to recursively update the probability density of the robot's position (the belief) using motion and perception information. Then, we represent the posterior probability density of the robot's pose with a set of discrete points in the configuration space of the robot [64]. The schema of the Adaptive Carlo Algorithm is indicated in Figure 24.



(a) Time step 1 (b) Time step 8

Figure 24. The Adaptive Carlo Algorithm [65]. It is shown that at Time step 8, there are significantly fewer particles than time step 1

Feature processing

Feature/Descriptor extraction

Descriptor extraction is a process in computer vision and image processing used to identify and extract features from an image or data that represent specific characteristics or properties. They help in analyzing and understanding the content of the image. In [1], The descriptor is a vector of values that represent the feature in a way that is invariant to scale, orientation and other transformations. They are coupled with navigation and position information to form a pathTile. To realize the extraction, the feature descriptor corresponding to each feature keypoint is computed.

Extracting the descriptor is a decoding operation. The descriptors are obtained from feature points. First, the image size and feature point position are normalized and (x,y) and K are used to represent the coordinates and number of feature points, respectively. Then, the actual positions of feature points on a certain channel in the tensor are obtained by inverse normalization, Finally, the output of a complete descriptor is obtained. The schema of descriptor extraction is indicated in Figure 25.

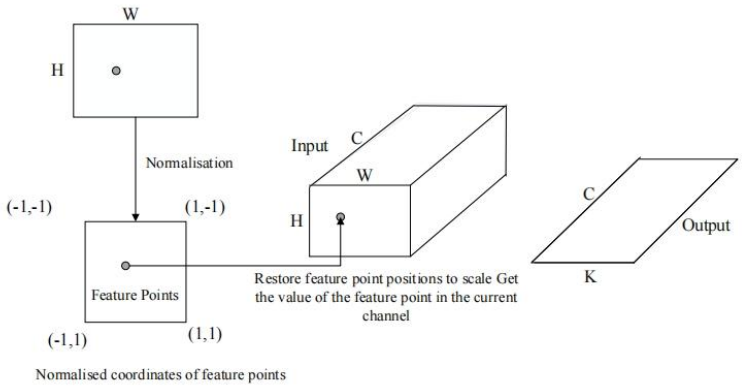


Figure 25. Descriptor extraction [66].

Moore-Neighbor contour detection algorithm

Moore-Neighbor contour detection algorithm is used to find objects' contour in binary image due to its simplicity and it is also reasonably robust. It identifies the

boundary of a shape by navigating through its pixels and using the 8-connected neighborhood to determine the direction of traversal, which makes the object detection and feature extraction realizable [67]. To better indicate the navigation process, the principle of the algorithm is indicated in Figure 26.

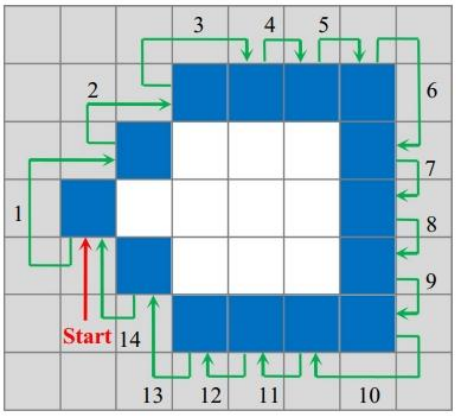


Figure 26. Principle of Moore's neighbour contour tracing algorithm [68].

This algorithm is significant in image processing as it allows for the extraction of shape features, enabling further analysis and recognition tasks. Compared to other methods, this method provides reliable and accurate localization results for AUV navigation and underwater work.

Feature/Descriptor Matching and Filtering

In descriptor matching, the key points are detected within the images, and then descriptors are calculated for each key point. A relevance score is then calculated between the query image descriptors and the reference image descriptors. In descriptor filtering, relevant approaches are applied (i.e. mean-shift clustering algorithm) to remove obvious mismatches [69]. For example, in [1], After the descriptor extraction and during the localization phase, the most recently collected pathTile descriptors are matched against the reference database collected during the training phase. Descriptor matches are further filtered by comparing the query and reference key points' sizes and angles. If the absolute difference between the size and angle of a query and reference key point exceeds a specified threshold, the match is rejected.

In reality, due to the fact that some parts of an image are similar, even with advanced descriptor matching, there are many descriptor mismatches after matching, which is shown in Figure 27. Therefore, descriptor filtering is necessary. Mousavi, V. et al. [69] carried out mismatch removal, for example, symmetry analysis, to filter out repetitive patterns in each image.

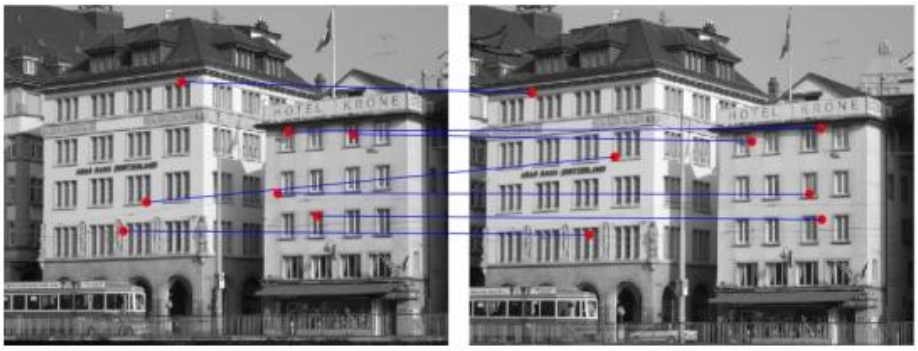


Figure 27. Example of mismatches occurred after the image matching [71].

Binary Classification Performance Evaluation

Binary classification performance evaluation involves assessing how well a binary classification model performs in distinguishing between two classes (commonly labeled as positive and negative). This evaluation is crucial for understanding the model's strengths, weaknesses, and overall effectiveness.

In [1], MCC is calculated to evaluate the performance of the prediction, it ranges from -1 to 1. A value of 1 represents a perfect prediction; a value of 0 is no better than random prediction; and a value of -1 represents a complete opposite prediction [1]:

$$\mathrm{MCC} = \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{11}$$

Where:

- True Positives (TP): Instances correctly predicted as positive.
- True Negatives (TN): Instances correctly predicted as negative.
- False Positives (FP): Instances incorrectly predicted as positive (Type I error).
- False Negatives (FN): Instances incorrectly predicted as negative (Type II error).

Self-Similar Landmarks (SSL)

The self-similar landmark is exactly or approximately similar to part of itself. In the underwater detection scenario, it is expected that: 1) a self-similar feature at long distances; 2) The self-similarity of the landmark attracts a robot until it approaches close to the landmark [70].

What is more, considering the complex underwater environment. For example, the underwater vehicle may be susceptible to rotation, the SSL should be rotation invariant. The detection may also be influenced by underwater turbidity. Therefore, the SSL should be immune to small blurs, which can increase the robustness of detection [71]. The SSL is shown in Figure 28.

Figure 28. A modified SSL that is rotationally invariant [71].

1.4.2. State-of-the-art

Underwater identification

Underwater identification aims to classify and recognize the category of underwater object, which is typically carried out before the localization. Relevant approaches can be classified into image processing, classification and deep learning technique.

Image processing

Image processing techniques aim to carry out some necessary tasks on the image to make it prepared for the classification, for example, image enhancement, feature extraction such as image segmentation and region of interest (ROI).

Image processing is an important technique for computer vision, it has been widely applied in marine detection, food evaluation, medical analysis [72, 73]

Some of the most relevant progress is in [74, 75, 76, 77, 78, 79, 80, 81]

Bosse, S. et al. [79] propose and evaluated a hybrid approach with segmented classification using small-scaled CNN classifiers. There are four steps for realizing identification. Firstly, image segmentation is carried out, where images are segmented into equally sized sub-images. Secondly, there comes the ROI Classification, where the CNN carries out parallel prediction of the image segment class. Thirdly, the system performs training and labelling, where each image was labelled manually by adding relevant and strong ROI polygons to each image. Finally, the Mean Bounding Box Algorithm realizes reconstruction of labelled ROIs, which filters wrong spurious segment classifications and represents the best and most accurate matching ROI. The prediction results for the training and test data do not differ significantly and show similar high statistical measures.

Chen, Z. et al. [80] propose an underwater object detection method using monocular vision sensors. The detection can be divided into 4 steps, In the first step, various features including intensity, color, and transmission are extracted from the raw underwater images. The second step involves global contrast calculation, which is realized through light transmission estimation. This transmission information is combined with the color and intensity features, which can facilitate the execution of ROI. The third step is ROI, this is because this method can emphasize the region that we will carry out the detection and mitigate low contrast issues. Finally, the extracted

ROI is filtered and corrected by the image segmentation method, producing the results of the underwater object detection. The classification of the detection result can be realized using machine learning methods such as SVM, Random Forest and K-Nearest Neighbors [82]. It is obtained that the system is demonstrated to be robust in underwater environments. The detection procedure is shown in Figure 29.

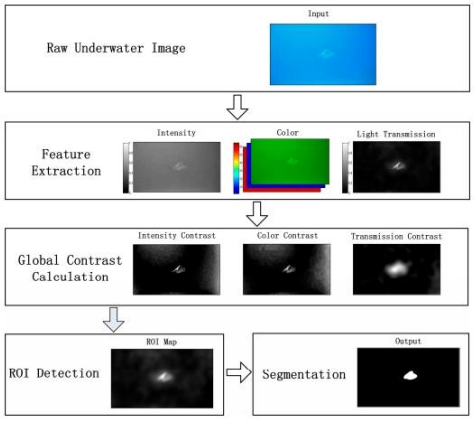


Figure 29 The detection procedure in [80].

Jian, M. et al. [81] provide some insightful descriptions about the algorithms of underwater identification. The first is Underwater Image Enhancement and Underwater Image Restoration/Recovery, then, there comes the Underwater Image Segmentation. In the next step, it is the Underwater Image Saliency Detection. Finally, the identification can be realized through machine learning methods. With the four methods combined, the object classification can be realized.

Classification

Classification is mainly realized by traditional machine learning methods. Transfer learning method is also included because it provides sufficient training data for machine learning, who often lacks data for training, especially deep learning. Traditional machine learning methods appear to be classical methods for object classification and identification.

Some of the most relevant progress is in [83, 84, 85, 86, 87, 88, 89, 90].

Jian, M. et al. [86] outlined some traditional machine learning methods used in underwater object detection. These methods involve manually extracting and combining traditional artificial features, such as texture, shape, color, and motion of targets, which are then used in conjunction with classical machine learning algorithms to achieve underwater object detection. The author points out that traditional underwater object detection methods rely on manual feature extraction, which is time-consuming and lacks robustness compared to deep learning methods.

Langner, F. et al. [87] developed a high-resolution sonar sensor for reliable object detection, classification and identification. There are four necessary steps: The first is preprocessing, which includes normalization, corrections for distortions; the second is screening, where the Region of Interest (ROI) are identified; the third is false alarms filtering, where false alarms are reduced by applying iterative fuzzy segmentation.

Finally, the classification process involves implementing a 2D cross-correlation with object templates and using the existence of parallel lines to identify the object shadow contour. The result shows that the approach can find the object. The classification procedure of this method is shown in Figure 30.

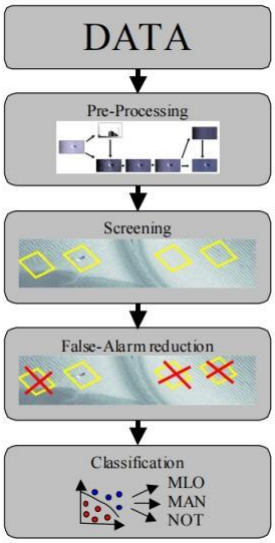


Figure 30. The classification procedure in [87].
(MLO – mine like object, MAN – man-made object, NOT – no target)

Dos Santos, M. et al. [88] presented an underwater object classification pipeline applied in acoustic images acquired by Forward-Looking Sonar (FLS). The process involves Image enhancement, Image segmentation, Segment description, and segment classification. In the segment description, some relevant information about the segment is calculated, including the height, width, segment area, perimeter etc. The classification methods include Support Vector Machine (SVM), Random Forest and K-Nearest Neighbors.

Luo, X. et al. [89] provided a review of current Underwater Acoustic Target Recognition (UATR) methods based on machine learning, for example, SVM, Gaussian Mixed Model (GMM). They point out that the traditional machine learning methods are based on traditional statistical methods to establish probability and statistical models for analysis and prediction. They are relatively simple and have easy-to-understand parameters. Traditional machine learning models can achieve good results on small datasets and are less prone to overfitting. To solve the problem of insufficient data, transfer learning is proposed. It first trains a network model on a large and related dataset called the source domain and then uses a small target domain dataset to fine-tune the parameters to make the network model adapt to the new task requirements.

Deep learning

Deep learning is a popular classification technique, which combines the function of image processing (i.e. feature extraction) and classification. Thanks to the significant development in neural networks and their strong classification ability, it can be said as the most popular method in image classification and identification field.

It has been applied in biomedicine, health management, and material [91, 92, 93].

Some of the most relevant progress is in [94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104].

Xu, S. et al. [99] provide a comprehensive review of techniques for the underwater object detection based on deep learning, moreover, they explored the internal relationship between underwater image enhancement and object detection. They mainly divide their paper into 3 parts, where datasets, image enhancement and object detection are described respectively. Different from previous views, they point out that underwater image enhancement does not directly lead to a higher detection accuracy. From Table 1, it can be found that after image enhancement, the mean average precision (AP) did not always increase. The AP on different image enhancement methods is shown in Table 2.

Table 2. The AP on different image enhancement methods [99]. The result indicates that underwater image enhancement does not directly result in higher accuracy

Method	Backbone	Origin	CLAHE	ICM	MSRCR	Fusion	S-U	Deblur
Two/Multi-stage Detector:								
Faster R-CNN [74]	R18	**50.0**	49.5	49.0	47.4	46.4	49.8	49.0
	R50	**53.8**	53.1	52.8	51.4	50.3	53.6	53.4
	R101	54.0	53.4	52.9	51.5	50.1	**54.4**	52.8
Cascade R-CNN [80]	R18	**52.4**	52.1	51.6	50.2	49.0	52.4	51.2
	R50	**55.7**	55.0	54.0	53.1	51.7	55.3	54.2
	R101	55.1	54.5	54.4	53.1	51.9	**55.4**	54.6
Dynamic R-CNN [86]	R18	**50.6**	50.0	49.4	47.7	47.2	50.3	49.5
	R50	**54.4**	53.6	53.0	51.4	50.2	53.6	53.4
	R101	53.8	53.3	53.1	51.7	50.4	**54.0**	53.1
Libra R-CNN [84]	R18	**50.2**	49.4	49.2	47.8	46.5	49.5	49.3
	R50	**54.2**	53.5	52.8	51.6	50.6	53.8	53.1
	R101	**54.4**	53.1	52.8	51.2	50.2	53.6	53.3
One-stage Detector:								
RetinaNet [87]	R18	**45.1**	43.4	43.1	41.8	41.0	43.6	43.7
	R50	**49.3**	48.9	48.4	47.2	45.1	49.2	48.4
	R101	49.6	48.9	48.7	47.4	46.2	**49.7**	49.1
FCOS [83]	R18	**50.5**	49.6	49.6	47.1	46.5	49.8	49.3
	R50	**54.6**	53.5	53.2	51.4	50.3	54.1	53.5
	R101	**54.4**	53.8	53.2	51.4	51.1	54.1	53.6
TOOD [89]	R18	**54.6**	54.0	53.4	51.2	50.7	54.3	52.9
	R50	**58.6**	57.6	57.1	55.1	54.2	57.9	57.3
	R101	**57.8**	57.5	57.0	55.3	54.2	57.7	57.0
ATSS [88]	R18	**54.2**	53.0	52.7	50.7	49.9	53.6	53.0
	R50	**57.6**	57.1	56.3	54.4	53.5	57.6	56.6
	R101	**57.6**	56.7	56.4	54.7	53.8	57.1	56.2
Underwater Object Detector:								
RoIAttn [126]	R18	**51.2**	50.8	50.7	48.6	47.8	50.8	50.1
	R50	-	-	-	-	-	-	-
	R101	-	-	-	-	-	-	-
Boosting R-CNN [127]	R18	**54.1**	53.6	53.1	51.6	50.4	53.9	53.1
	R50	**58.1**	56.6	56.1	54.8	53.3	57.8	56.6
	R101	57.3	56.4	55.9	54.6	53.4	**57.4**	55.7
Boosting R-CNN* [127]	R18	**53.9**	53.4	53.1	51.6	50.0	53.6	53.0
	R50	**57.9**	56.7	56.2	54.7	53.6	57.7	56.2
	R101	56.9	56.4	55.9	54.5	53.1	**57.1**	55.7

Chen, L. et al. [4] comprehensively study the AI-based Underwater Object Detection (UOD) and provide a description for the traditional machine learning-based methods and deep-learning-based methods. For deep learning methods, they can be classified into Transferring Generic Object Detection Frameworks and Specially Designed Underwater Object Detection Frameworks. For the former framework, it can be divided into one-stage detectors and two-stage detectors. For the latter framework, novel frameworks or algorithms are specially designed to tackle the unique challenges of UOD. It is noteworthy that traditional machine learning methods have recently fallen behind advanced deep learning techniques because of their limited accuracy.

Er, M. J. [100] offered comprehensive challenges and corresponding solutions of deep learning-based underwater marine object detection techniques based on vision sensors. Firstly, image quality degradation is a main challenge. This can be solved

through image enhancement. Secondly, small object detection is also a problem. To deal with it, methods like feature fusion and contextual information methods are proposed. Then, Generalization in Underwater Object Detection can be the other drawback, which can cause domain drift and thus dramatically degrade detection performance. To cope with the problem, methods like data augmentation are proposed. Finally, real-time detection is a non-negligible factor. To realize it, the one-stage detector is preferred rather than the two-stage detector. The experimental results show that after the methods which deal with challenges are applied, an improved performance of the system can be detected. Small instances in underwater object detection are shown in Figure 31.

Figure 31. Small instances in the underwater object detection [100]. (a) Fish (b) benthic organisms

Guntha, P. et al. [101] offered a systematic review of recent advancements in this field, covering both image enhancement and object detection techniques. For image enhancement, it aims to mitigate poor visibility caused by scattering and absorption of light underwater. For example, Cheng, N. et al. [102] utilized image enhancement subnetwork to enhance object detection by minimizing pixel domain error through a combination of background light and medium transmission. In the next step, object detection is carried out, which is used to classify and identify the object. For example,

Gašparović, B. et al. [105] utilized YOLO and Faster R-CNN series for the underwater object detection. The experimental result shows that YoloV7 has an excellent overall performance considering the processing speed and the precision.

Jian, M. et al. [106] presented traditional methods and advanced technologies based on deep learning used in underwater object detection. Moreover, it conducts a comprehensive study of seven representative datasets used in underwater object detection missions. The traditional methods involve extracting and combining traditional artificial features, such as texture, shape, color, and motion of targets, which are then used in conjunction with classical machine learning algorithms to achieve underwater object detection. Deep learning methods, such as convolutional neural networks (CNNs), have gained widespread popularity because of their ability to automatically extract and classify features from underwater images, thus leading to improved accuracy in detection and recognition tasks. Finally, the underwater datasets are shown, which are used to satisfy the high image data demand caused by complex and blurry underwater environments. Diagram of an example frame from the Brackish dataset category is indicated in Figure 32.

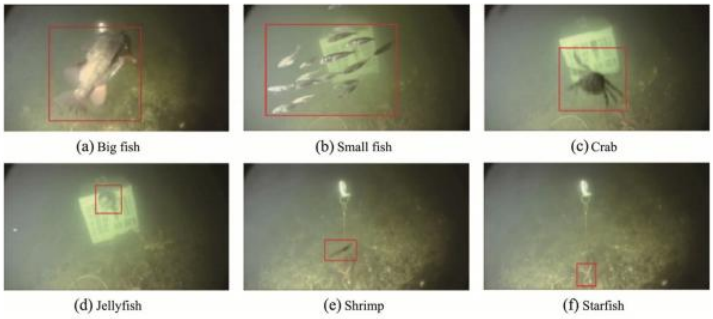


Figure 32. Diagram of an example frame from the Brackish dataset category [106]. Brackish is an underwater dataset that contains more than 14000 images

To overcome the shortcomings of the traditional manual detection of underwater targets in side-scan sonar (SSS) images, Yu, Y. et al. [6] propose a real-time automatic target recognition (ATR) method. It can be divided into 3 steps, which are image preprocessing, image sampling and object detection. Image preprocessing aims to improve the contrast. Image sampling is carried out to prevent the underwater target from being divided into two images. Finally, object detection utilizes TR－YOLOv5s algorithm to classify and a transformer to enhance the key information. TR-YOLOv5s is a modified deep learning method, which adds a transformer module to adapt to sonar imaging, while initially it is for optical imaging. The result shows that compared with object detection after SSS measurement, the proposed real-time detection is very efficient for quick object detection on the spot.

Zhu, P. et al. [103] present an automatic target recognition (ATR) approach for sonar onboard unmanned underwater vehicles (UUVs). In this approach, image preprocessing is applied to enhance the contrast. Then, target features are extracted by a convolutional neural network (CNN) operating on sonar images. Finally, the extracted features are classified by a support vector machine (SVM) that is trained based on manually labeled data. The experimental results show that the features extracted by the CNN method can describe the target objects in the sonar images better than the LBP [107] and HOG [108] features. The training architecture of SVM is shown in Figure 33.

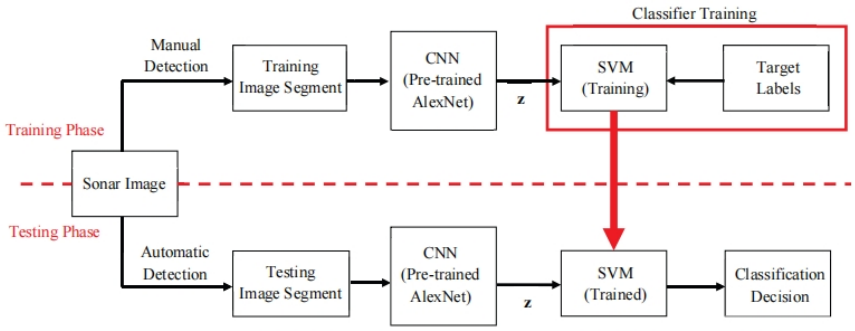


Figure 33. Training architecture of SVM [103]. The training phase should be completed before the testing phase is carried out

Ge, H. et al. [104] proposed a deep-learning-based underwater target detection system that can effectively solve the problem of underwater optical image target detection and recognition. The functioning procedure can be divided into: Image enhancement, Object detection and Data network parameter transfer. The image enhancement proposes a method based on the generative adversarial network for aquatic image degradation. In object detection, YOLOv3-Based Lightweight Detection Model is posed to save energy and release more storage resources. In Data network parameter transfer, a method based on transfer learning is proposed. The experimental result shows that the proposed model outperformed other similar models.

1.4.3. Underwater localization

Localization of underwater structures is a common procedure in fields such as underwater surveillance, operations, maintenance, and measurement [109]. The localization is carried out when the object is already identified.

SLAM

Simultaneous Localization and Mapping (SLAM) is a popular technique for localization of the underwater structure, which is usually realized by simultaneously constructing or updating a map while localizing itself. The generation of the map provides the position information of the detected object.

It is mostly applied in domains such as environmental monitoring and military operations [110].

Some of the most relevant progress is in [111, 112, 113, 114, 115, 116, 117].

Wang, X. et al. [111] systematically introduce the underwater Simultaneous Localization and Mapping (SLAM) technologies, where framework, relevant sensors and techniques for SLAM are introduced. The key points for the paper are the sensor information and techniques for SLAM. It is shown that the SLAM requires various types of sensors to improve accuracy and robustness. For example, the camera, sonar sensors and lidar sensors. The system structure of SVIn2, which fuses several sensors to achieve SLAM, is indicated in Figure 34. In addition, there are relevant methods which can be used to realize sensor fusion in integrated navigation systems. For example, Kalman filter based methods and graph-based optimization methods. They provide a clear overview of equipment and technology requirements for SLAM. However, they do not pose any experiments to demonstrate a certain result.

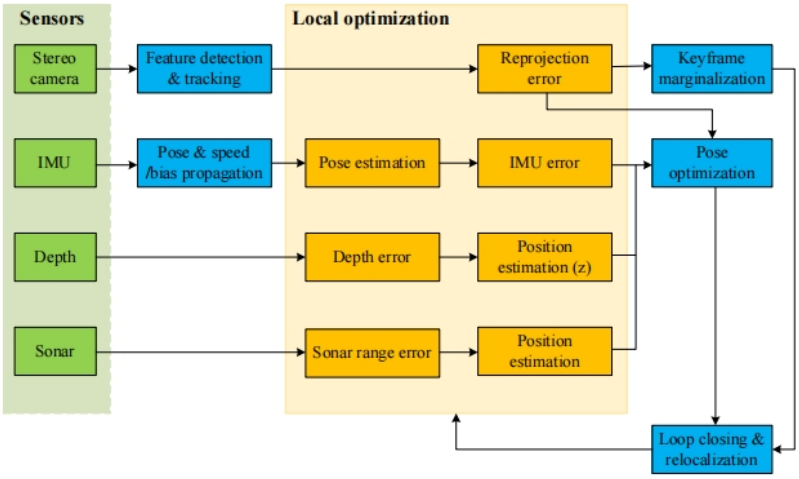


Figure 34. System structure of SVIn2: an underwater SLAM system using sonar, visual, inertial, and depth sensors [111].

Rahman, S. [118] provides a comprehensive study of the visual SLAM. In the paper, he formulates a novel SLAM system, where acoustic (mechanical scanning profiling sonar), visual (stereo camera), inertial (linear accelerations and angular velocities), and depth data is fused together to map different underwater structures. Hence, the sonar, visual, inertial and depth sensor are applied. According to the results obtained, the SLAM system can generate a track with little error. However, it did not work in reality because of the poor light. Indicating that light is an important factor for the SLAM system. The block diagram of the system is indicated in Figure 50.

Viset, F. et al. propose an approach to magnetic field SLAM that is faster and requires less storage compared to the approach proposed in [119], where the magnetic field SLAM was proposed to compensate for odometry drift when there is access to magnetic field. The Extended Kalman Filter is applied rather than the Particle Filter [120]. The Extended Kalman Filter indicates a higher computational efficiency but is less accurate. The simulation demonstrates the drift-compensating abilities of the EKF-SLAM algorithm on a set of data.

Visual localization

Visual localization is a typical and common method in underwater localization. It is a popular technique for short-range, precise localization.

They are widely applied in unmanned aerial and land vehicles, marine detection, humanoid robot [121].

Some of the most relevant progress is in [122, 123, 124, 125, 126, 127, 2, 128, 129].

Nuske, S. et al. [127] developed a vision based localization method that uses a 3D model of the structure to be inspected. The localization process is divided into 3 steps, putting them in order, it is: Synthetic Image Generation, Particle Filter Localization and Gradient-domain Image Matching. In the first step, synthetic images are generated from a polygon mesh of the structure. The mesh includes the surface normal, diffuse and specular reflectance properties. Then, the synthetic image generated from the particles nearest the correct pose will be selected as the best match with the real image. Finally,

the comparison process between the camera image and a synthetic image is carried out to provide a likelihood measure for the set of particles in the filter. Results show that the system can localize the vehicle in challenging image sequences where the light source is constantly moving and illuminating the scene non-uniformly. However, the system makes a mistake when one of the columns of the structure is hindered by another. The experiment setup is shown in Figure 35.

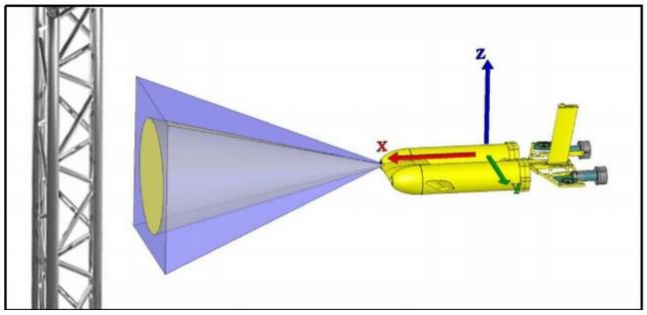


Figure 35. The experiment setup [127]. The AUV has a forward facing camera with a field-of-view depicted in the figure by pyramid viewing volume. The spotlight is also facing forward and partially illuminates the field-of-view

Xanthidis, M. et al. [2] proposed a multi-robot mapping framework that utilizes two types of robots, which are both underwater vehicles (AUVs). The first robot is termed proximal observers, which will be operating close to the underwater structure generating a dense vision-based 3D reconstruction of the observed surface. The rest of the robots, termed distal observers, will operate further out optimizing the global picture of the underwater structure and the pose of the proximal observers with respect to the structure. Finally, the pose can be obtained by Robust State Estimation. The experimental results show that the proposed switching estimator managed to keep track throughout the trajectory with the lowest root mean square error (RMSE). This solves the problem of a single AUV's loss of tracking when no unique visual feature is present. The distal observers detecting the proximal observer and the structure is indicated in Figure 36.

Figure 36. The distal observer is detecting the proximal observer and the structure [127].

Zhong, L. et al. [129] developed a binocular localization method for AUV docking. The process can be divided into imaging processing and binocular vision algorithm.

The former step is to prepare the image for the binocular vision algorithm, which includes image filtering, feature extraction and image segmentation. The later step is to realize the localization of a vessel, it first obtains the position of three matching points in the AUV coordinate and dock coordinate, then it calculates the transformation matrix. Hence, the position and attitude of AUV can be obtained. A verification experiment was conducted in the laboratory pool using a ship model to evaluate the feasibility of the entire system. The ship model can achieve docking no matter whether it starts at the right side or the left side of the docking station.

Qin, J. et al. [130] provided a literature overview of the vision based navigation and positioning of autonomous UUVs. The paper classifies the vision-based localization into Geometry-Based Methods and Deep Learning-based Methods. Among all Geometry-Based Methods, the Vision-Based Multi-Sensor Fusion Method, which fuses data of camera, IMU or other sensors like sonar, indicates a higher accuracy and robustness. The Deep Learning-based Methods can automatically discover task-relevant features using highly expressive neural networks, which also makes them better suited to more scenarios. Relevant algorithms include CNN, Recurrent Neural Network (RNN) [131] and GAN.

Visual Marker

Visual marker is a popular passive localization method that provides high-precision results within short detection range.

It is widely applied in localization for various scenarios.

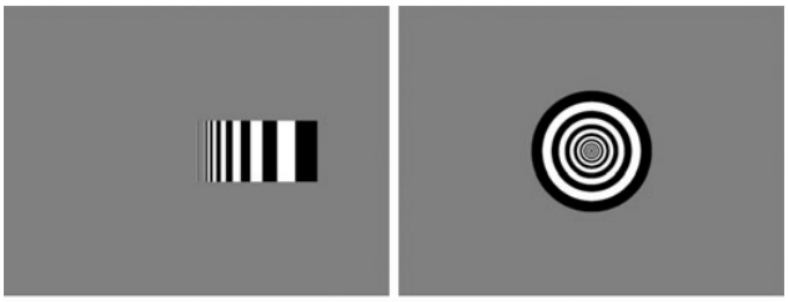
Some of the most relevant progress is in [132, 133, 134, 135, 136, 137, 71].

Negre, A. et al. [71] posed a method for vision-based underwater target identification and localization, where the Self-Similar Landmarks (SSL) plays an important role for the target identification. To successfully carry out the process, the landmark should be specially designed. The detection algorithm, which applies matching function on every pixel of the image, is also considered. Finally, the pose and range estimation are carried out, whose result depends on the number of visible landmarks. The experiment result shows that the AUV can maintain a 2m distance from the target object and it can identify the target within this distance. This research poses a novel underwater detection method. However, the way of attaching the landmark to the object of interest would be a challenge. The landmark that is improved and described in the paper is indicated in Figure 37.

Figure 37. Circular 3-pattern SSL target which is rotationally invariant [138].

Wei, Q. et al. [137] posed an underwater localization method based on enhanced visual markers. The visual marker incorporates encoded features and is carefully designed to yield greater precision and lower variability. Then, the Underwater Optical Image Restoration Model is applied to facilitate the identification and localization of underwater objects by enhancing the quality of underwater images. Finally, the underwater visual localization algorithm, which integrates image restoration, feature detection, geometric encoding value analysis, and pose estimation, is applied to determine the location. According to the experimental results, this method consistently outperforms analogous techniques in all the evaluated metrics, however, if the visual markers are obscured, it will be challenging to obtain accurate results.

Negre, A. et al. [71] evaluates experimentally the use of self-similar landmarks (SSL) to realize target localization and identification, which enables the AUV to realize short-range homing and docking operations. In the first step, a modified SSL is designed to ensure that the SSL is rotationally invariant and can improve robustness to noise and motion blur. During the operation, the target detection is first carried out, where the position of the target is determined by the detection algorithm. Then, the pose and range of the target with respect to the camera is determined depending on the number of visible landmarks. Hence, the localization and identification can be realized. According to the experiment, the landmark posed by the method is exceptionally robust with very little sensitivity to camera model, distortion, and observation range, making it a good selection for short-range homing and docking operations. The comparison between the original marker and the modified marker is shown in Figure 38.



(a) Original marker (b) Modified marker

Figure 38. Comparison between the original marker and the modified marker [71]. For (a), It is not robust to transformations like motion blur. While for (b), it can overcome the problem

1.4.4. Knowledge gap

Image enhancement

Underwater object identification provides categorical information about detected structures and is often preceded by an image enhancement step to improve image quality. Image enhancement has been widely used as a preprocessing technique to boost classification performance, with many studies reporting improvements in detection outcomes.

Despite its widespread application, the direct influence of image enhancement on

object identification accuracy remains insufficiently explored. The relationship between enhancement techniques and identification performance in underwater environments is still unclear.

Underwater identification

Deep learning has emerged as a leading approach for image identification and classification, with one-stage and two-stage detectors representing the primary architectures. Traditionally, one-stage detectors are considered faster but less accurate than two-stage detectors. The training process equips the identification model for its recognition tasks, and typically, an increase in training iterations correlates with improved model accuracy.

However, recent studies rarely elaborate on the operational mechanisms of one-stage and two-stage detectors, which hinders their practical implementation. While it is well-established that extended training enhances model accuracy, the optimal number of training iterations required to develop a sufficiently precise identification model remains unclear.

Underwater localization

Localization of underwater structures is a critical component for effective underwater operations. Current techniques predominantly rely on visual and acoustic modalities, with many studies focusing on offline or non-real-time implementations. Recently, deep learning has gained traction due to its promising classification accuracy and potential applicability to underwater localization tasks. Additionally, magnetic-based localization approaches have been explored as alternative solutions.

Despite these advancements, several challenges persist. Real-time visual-based localization has been insufficiently explored, and existing studies often lack the implementation details of visual methods. In addition, while deep learning holds significant promise for underwater localization, its application remains limited in literature. Furthermore, magnetic-based methods often encounter computational limitations, resulting in suboptimal accuracy and processing speeds.

1.5. Research questions

Before initiating the research, the central research question and its corresponding sub-questions will be clearly defined. Based on the literature, visual localization emerges as a crucial technique for underwater positioning. Furthermore, object identification in underwater environments is predominantly achieved through machine learning and deep learning approaches, with image-based identification being a common focus across existing studies. These insights suggest that the research should concentrate on two core components: the use of underwater cameras and the application of machine learning or deep learning algorithms. The ultimate objective is to develop an integrated system that combines object identification and localization to enhance the functionality of underwater robotic operations. Accordingly, the main research question is formulated as follows:

How can underwater targets be localized, identified, and engaged using underwater robotics?

The sub-questions are listed as follows:

1. How to identify the underwater structure using the underwater camera?
2. How to localize the underwater structure using the underwater camera?
3. How to engage the target to realize the fixation between the target and the robotics?

In the rest of the report, the research tries to answer the main question and the sub-questions as clearly as possible.

1.6. Report structure

Chapter 1 of the graduation thesis is the introduction, which includes background and motivation, problem statement, underwater imaging, literature review, research questions and report structure.

Chapter 2 describes the background knowledge for the graduation thesis, which includes the description of Faster RCNN and identification false-prevention methods.

Chapter 3 presents the methodology for the graduation thesis, which includes image enhancement, methodologies for underwater identification and localization.

Chapter 4 discusses the experiment, which includes the experiment setup and experiment procedure.

Chapter 5 shows the experimental results for underwater identification and localization.

Finally, chapter 6 presents the conclusions and recommendations.

There are also contents that are attached to the Appendix, including Scientific Research Paper, Grabber mechanism, GUI (Graphic User Interface), Image enhancement results, Running environment requirements, Training dataset for deep learning and Alternative depth estimation algorithm, etc. They serve as supplements to the main report.

2. Background knowledge

To support the methodology presented in this study, it is essential to review relevant background knowledge, particularly in object detection algorithms. This section introduces two key components: the Faster RCNN algorithm and an identification filtering algorithm.

2.1. Faster RCNN

Faster R-CNN is a widely adopted deep learning algorithm for object detection, known for its balance between detection accuracy and computational efficiency. It belongs to the R-CNN (Region-based Convolutional Neural Network) family, which evolved through several stages of development: RCNN, Fast RCNN and Faster RCNN.

RCNN

The original RCNN algorithm begins by generating many region proposals (i.e., candidate bounding boxes) from an input image. Each proposal is then processed independently to determine whether it contains an object, which is computationally intensive. After filtering and refining the proposals, class labels and final bounding boxes are assigned. The workflow of RCNN is illustrated in Figure 39.

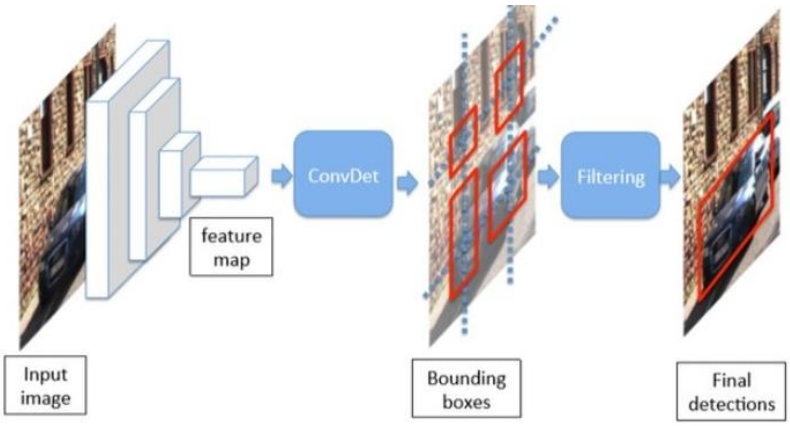


Figure 39. The RCNN series [139].

Fast RCNN

To address the high computational cost of RCNN, Fast RCNN improves efficiency by running the convolutional neural network only once per image. Regions of Interest (ROIs), which may contain objects, are extracted from the resulting feature map. This significantly reduces computation time and improves resource efficiency compared to RCNN.

Faster RCNN

Faster RCNN builds upon Fast RCNN by introducing a Region Proposal Network (RPN), which shares convolutional layers with the object detection network. The RPN receives the feature map as input and generates region proposals along with objectness scores. These proposals are then passed to the Fast R-CNN module for final

classification and bounding box refinement. This integrated architecture substantially accelerates detection while preserving high accuracy. The structure of Faster R-CNN is depicted in Figure 40.

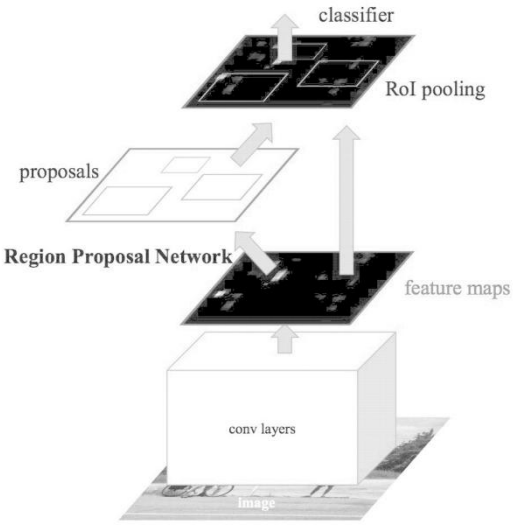


Figure 40. The Faster RCNN [140].

In practical applications, implementations of the RCNN series are available through open-source libraries. One of the most widely used is Detectron2, developed by Facebook AI Research. Detectron2 is a modular and flexible object detection framework that supports various models, including Faster R-CNN, RetinaNet, and Mask R-CNN. The official homepage of Detectron2 is shown in Figure 41.

Figure 41. The home page of Detectron2 [141].

Detectron2 provides a “Model Zoo” where both pre-trained and untrained models

can be accessed. These models can be easily integrated into local Python environments or Google Colab notebooks. With a properly prepared dataset, users can train models for specific tasks such as underwater object detection.

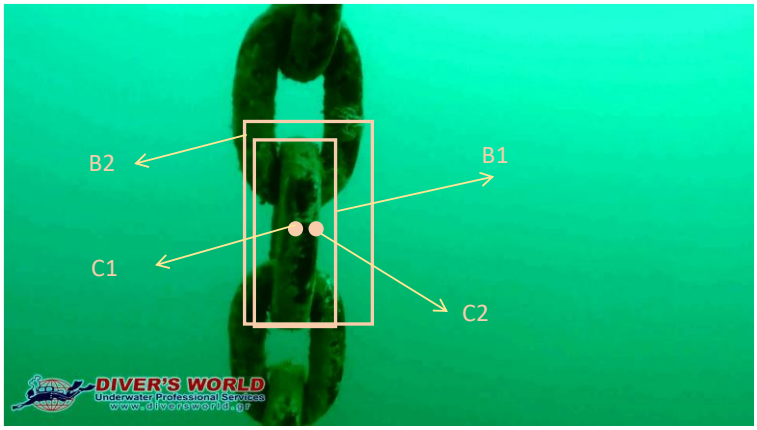
2.2. Identification false-prevention methods

To enhance detection accuracy during the chain labelling process, several false identification prevention techniques are employed. These methods are designed to reduce false positives and maintain consistent object identification across sequential frames. The primary strategies include Distance filtering, Abnormal object filtering, and Detected object recall mechanisms.

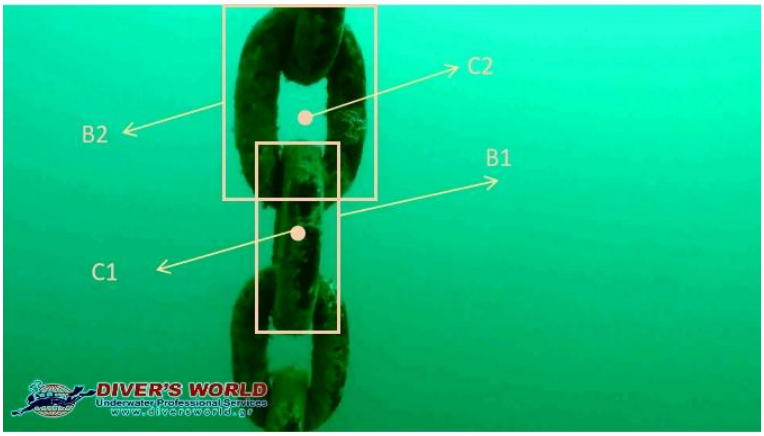
Distance filtering

Due to minor variations in the predicted positions of bounding boxes, the same object may be detected at slightly different locations across consecutive frames. This can lead to redundant or inconsistent labelling. To address this issue, distance filtering is applied to assess whether a newly detected bounding box corresponds to a previously identified object.

At the beginning of the detection sequence, the first bounding box is always accepted as a valid detection, and the coordinates of its center are recorded. For each subsequent bounding box, its center position is compared to the centers of existing tracked objects. If the Euclidean distance between the new center and any existing center exceeds a predefined threshold, the bounding box is classified as representing a new object. Conversely, if the distance is within the threshold, it is considered a continuation of the same object. This concept is illustrated in Figure 42, where B1 represents Bounding box 1, C1 represents Center 1, B2, C2, B3, C3 are similar to that of B1 and C1.



(a) Case 1: Two bounding boxes are regarded as one object.



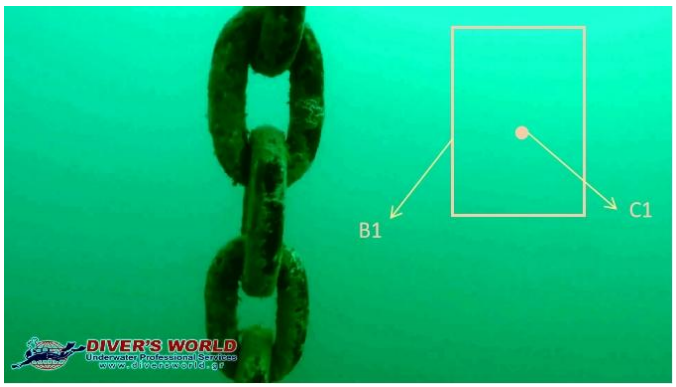
(b) Case 2: Two bounding boxes are regarded as different objects.

Figure 42. Diagram of the distance filtering.

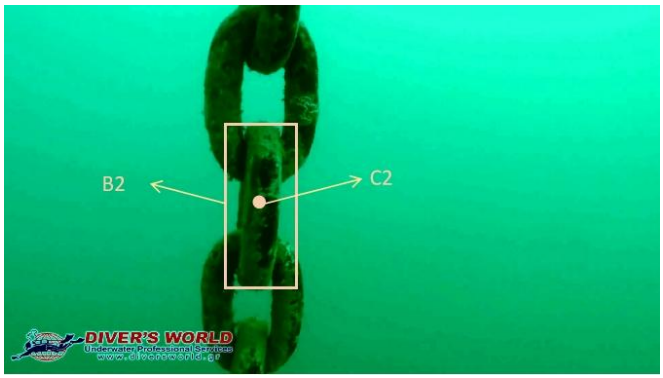
In Figure 42 (a), the centers are close together, leading to the interpretation of a single object. In contrast, Figure 42 (b) shows centers far apart, triggering the creation of a new object ID.

Abnormal object filtering

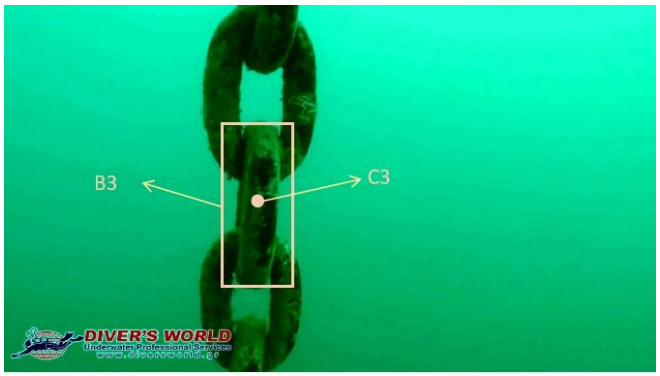
Misidentification can occur due to environmental noise or insufficient model training, resulting in false detections. These short-lived errors can increase the object count incorrectly. Abnormal object filtering mitigates this by ignoring objects that appear fewer times than a defined threshold. For instance, if a bounding box appears in fewer than three frames, it is considered an anomaly and excluded from the final object count. This method is demonstrated in Figure 43.



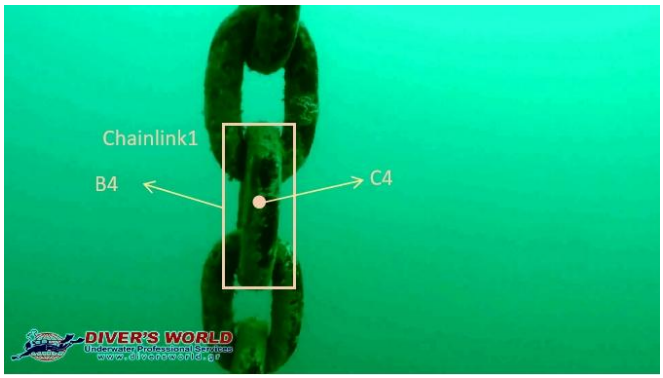
(a) Frame 1



(b) Frame 2



(c) Frame 3



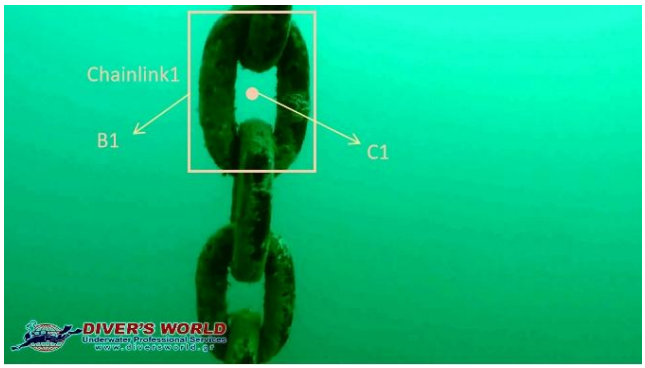
(d) Frame 4

Figure 43. The diagram of abnormal object filtering.

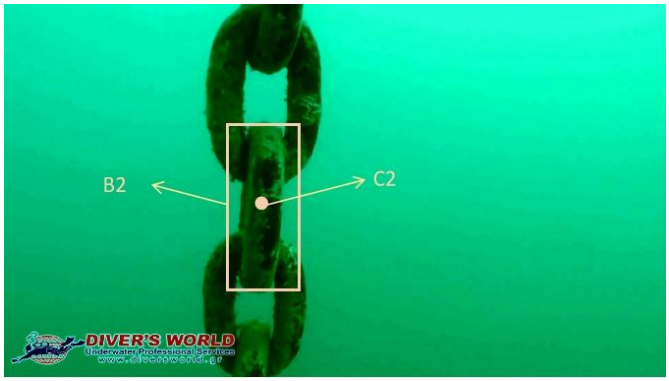
As shown, the incorrect detection appears only once across four frames. With a threshold of three appearances, the incorrect detection is discarded, while the correct object is retained.

Detected object recalling

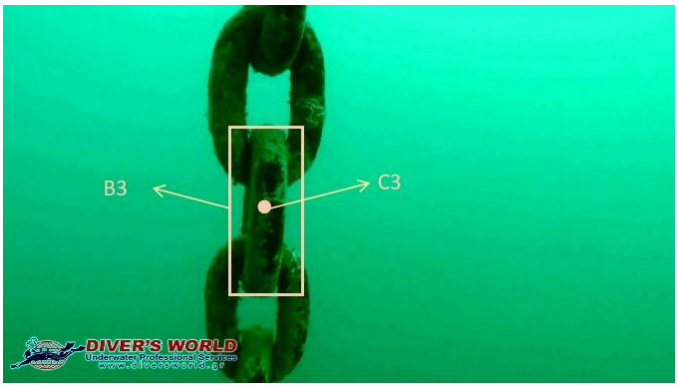
Sometimes, an object previously identified may temporarily disappear due to occlusion or failure detection. When the object reappears, it may receive a new ID, leading to overcounting. Detected object recalling addresses this by matching new detections to previously identified objects based on spatial proximity. If the center of a newly detected bounding box lies within a certain distance of a previously known object center, the previous ID is recalled and reassigned. Recalling is time-limited and is "forgotten" if the object remains undetected for too long. The process which explains the detected object recalling is indicated in Figure 44.



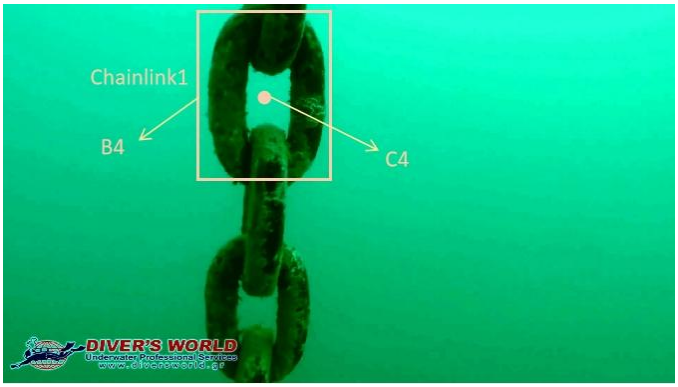
(a) Frame 1



(b) Frame 2



(c) Frame 3



(d) Frame 4

Figure 44. The diagram of detected object recalling.

As shown in the diagram, the numbered chain link is not detected in frames 2 and 3. However, in frame 4, when the center of the new bounding box is close to the previous center of the chain link, the system recalls the original number. This helps avoid counting the same chain link more than once and keeps the identification accurate.

3. Methodology

This section outlines the methodology employed to address the challenges of underwater image quality, object identification, and localization. The methodology comprises three main components: image enhancement, underwater identification, and underwater localization. Details about the running environment can be found in Appendix E.

3.1. Image enhancement

Underwater image enhancement is critical to improve the quality of image, including eliminating noise and color distortion, enhancing features of interest, weakening irrelevant background features. This preprocessing step aims to improve the performance of subsequent identification and localization tasks. In this paper, the IFM (Image formation model)-based image enhancement is applied, relevant image enhancement techniques can be found in this paper [142].

For the target underwater videos, the RGHS method is applied due to its superior enhancement quality as compared to other methods evaluated in [142]. Both CLAHE and RGHS use adaptive parameters to avoid global histogram stretching and preserve sharpness, but RGHS provides better dehazing effects. The image enhancement result by CLAHE and RGHS is shown in Figure 45.

(a) Before enhancement

(b) Enhanced by CLAHE

(c) Enhanced by RGHS

Figure 45. The image enhancement result.

Figure 45 compares the enhancement results using CLAHE and RGHS. While CLAHE improves contrast and visibility, RGHS yields clearer images with significantly better contrast. Accordingly, RGHS is used throughout this work for similar underwater scenarios. The underwater detection shown in Figure 46 also utilizes the RGHS method.

(a) Before enhancement

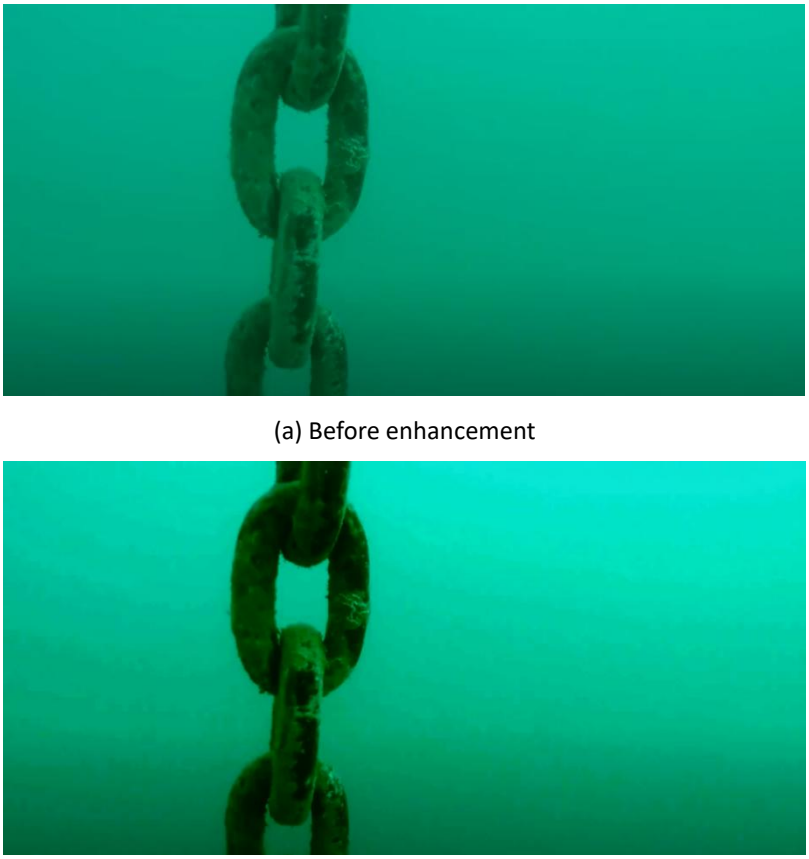
(b) After enhancement

Figure 46. A successful image enhancement realized by RGHS.

The results indicate an improved contrast as well after the enhancement, showing that the RGHS can be a highly effective method when the video to be enhanced has a

low contrast and visibility, which makes the harsh underwater identification easier to some extent. The results of other image enhancement methods are shown in Appendix D.

However, enhancement may not always result in improved identification accuracy, especially when the original image quality is already adequate. In such cases, enhancement can introduce unrealistic. The enhancement result that is unrealistic is indicated in Figure 47.



(a) Before enhancement

(b) After enhancement

Figure 47. The enhancement result that does not lead to better performance.

Therefore, it is advisable to select an enhancement technique that is tailored to the specific underwater environment.

An alternative method, FUnIE-GAN, a deep learning-based approach, was also evaluated. This method demonstrated suboptimal performance and imposed image size constraints, making it unsuitable for the application. The image enhancement effect of FUnIE-GAN is indicated in Figure 48.

(a) Before enhancement (b) After enhancement

Figure 48. The image enhancement effect of FUnIE-GAN

3.2. Underwater identification

Underwater identification forms the basis for localization by recognizing and classifying detected objects. In this study, a deep learning-based approach is employed, with a focus on chain link detection. The underwater identification includes object identification and object labelling.

3.2.1. Object identification

Deep learning algorithms generally outperform traditional machine learning in object identification tasks. These algorithms can be categorized into one-stage (e.g., YOLO series) and two-stage methods (e.g., RCNN series). The two-stage algorithm requires more computational power and is more precise. To achieve a higher precision, the two-stage Faster RCNN is adopted.

According to the literature review, object identification based on deep learning leads to a better performance compared to machine learning because deep learning methods can learn from their own errors, while machine learning often requires human intervention. Hence, object identification is realized through the deep learning algorithm.

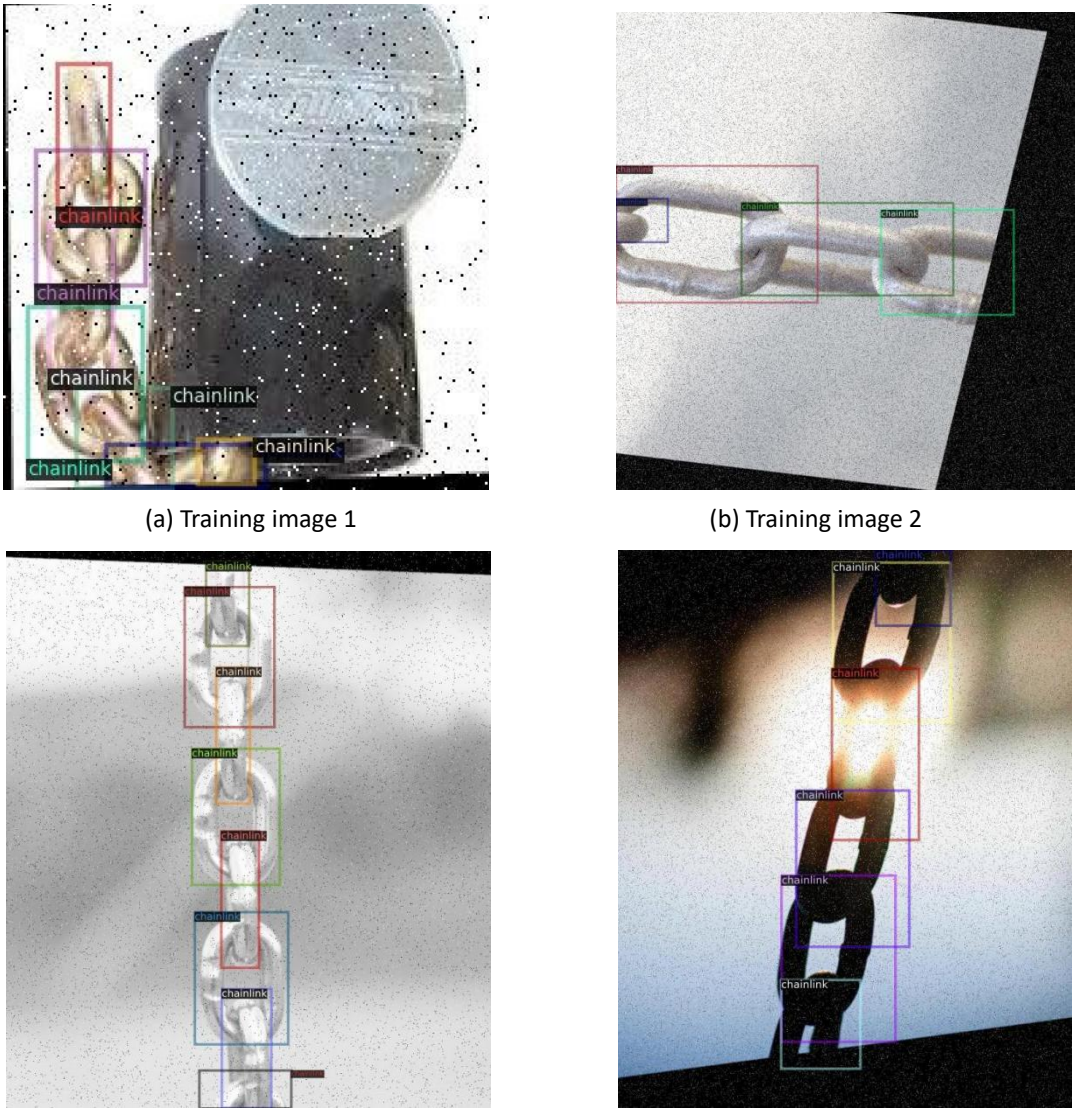
The deep learning algorithm for object identification can be classified into one-stage algorithm and two-stage algorithm. The two-stage algorithm requires more computational power and is more precise. The most common one is RCNN series. While the one-stage algorithm is the reverse of the case, the most common algorithm for it is YOLO series. To increase the identification precision, the two-stage algorithm, Faster RCNN is applied. The identification is carried out as the procedures shown below:

Training set acquisition

Please refer to Appendix F. Training dataset for deep learning

Training

In the model training process, it is assumed that the training dataset has already been downloaded. The training program then uses this data to train the identification model. Examples of images from the training dataset are shown in Figure 49.



(a) Training image 1 (b) Training image 2

(c) Training image 3 (d) Training image 4

Figure 49. Some images in training dataset.

Prior to the commencement of training, an appropriate identification model must be selected. In this study, models demonstrating higher precision are prioritized. Consequently, the 'faster_rcnn_X_101_32x8d_FPN_3x' model was chosen, as it exhibits the highest precision among the evaluated models, despite having comparatively lower training and inference speeds. A summary of the Faster RCNN algorithms is presented in Table 3.

Table 3. The overview of Faster RCNN algorithms on Dectron2 [143].

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	model id
R50-C4	1x	0.551	0.102	4.8	35.7	137257644
R50-DC5	1x	0.380	0.068	5.0	37.3	137847829
R50-FPN	1x	0.210	0.038	3.0	37.9	137257794
R50-C4	3x	0.543	0.104	4.8	38.4	137849393
R50-DC5	3x	0.378	0.070	5.0	39.0	137849425
R50-FPN	3x	0.209	0.038	3.0	40.2	137849458
R101-C4	3x	0.619	0.139	5.9	41.1	138204752
R101-DC5	3x	0.452	0.086	6.1	40.6	138204841
R101-FPN	3x	0.286	0.051	4.1	42.0	137851257
X101-FPN	3x	0.638	0.098	6.7	43.0	139173657

Several key parameters significantly impact the training process. The first is IMS_PER_BATCH (batch size), which defines the number of training samples processed in each iteration. Choosing an appropriate batch size is critical, as it influences training efficiency, computational resource utilization, and model convergence behavior. The second parameter, MAX_ITER, specifies the total number of training iterations. While increased iterations generally enhance model accuracy, excessive training may cause overfitting. In this study, MAX_ITER is set to 600, resulting in an accuracy exceeding 0.85 without signs of overfitting. Finally, ROI_HEADS. NUM_CLASSES determines the number of object categories to be identified; since this project focuses exclusively on chain link detection, it is set to 1. Additional training parameters are summarized in Table 4, and the learning rate curve is illustrated in Figure 50.

Table 4. The relevant training parameters for deep learning

Name	Value	Description
BASE_LR	0.00025	This represents the learning rate, it affects how much the model weights change in each step. A higher learning rate will cause the weights to be updated more aggressively
BATCH_SIZE_PER_IMAGE	128	This is a parameter that is used to sample a subset of proposals coming out of RPN to calculate cls and reg loss during training. The smaller the value, the faster the model is, but the precision may decrease.
NUM_WORKERS	2	This represents the number of processes that generate batches in parallel. A high enough number of workers ensures that CPU computations are efficiently managed

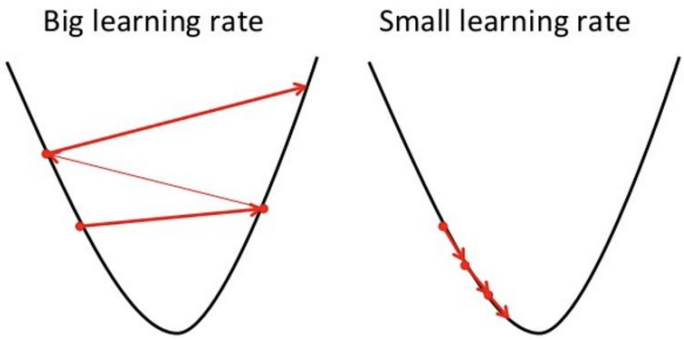


Figure 50. The learning rate [144]

Validation

Please refer to Appendix J. Validation

Testing

The testing procedure enables evaluation of the identification model's performance by applying it to input data not utilized during training. To process an entire video, it is first segmented into individual frames. Each frame is then sequentially analyzed by the identification model, which generates predictions for detected objects. After processing all frames, they are recombined into a video using a dedicated program. The output video displays object classification results along with associated confidence scores. To minimize false positives, a confidence threshold is applied to exclude predictions with low confidence levels. Figure 51 illustrates the consequences of selecting an inappropriate confidence threshold.

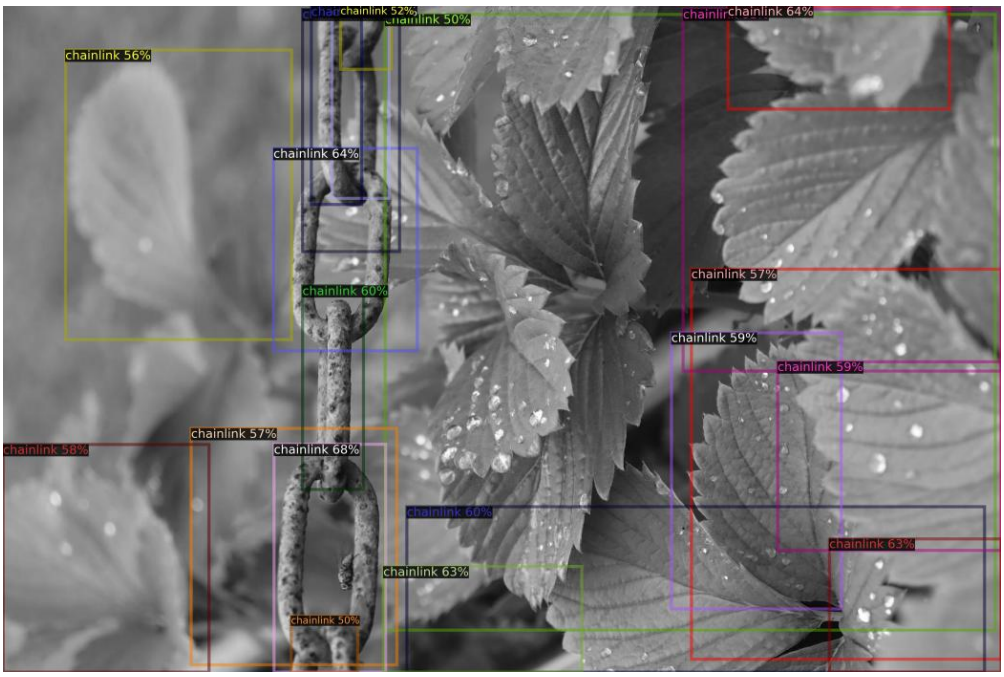
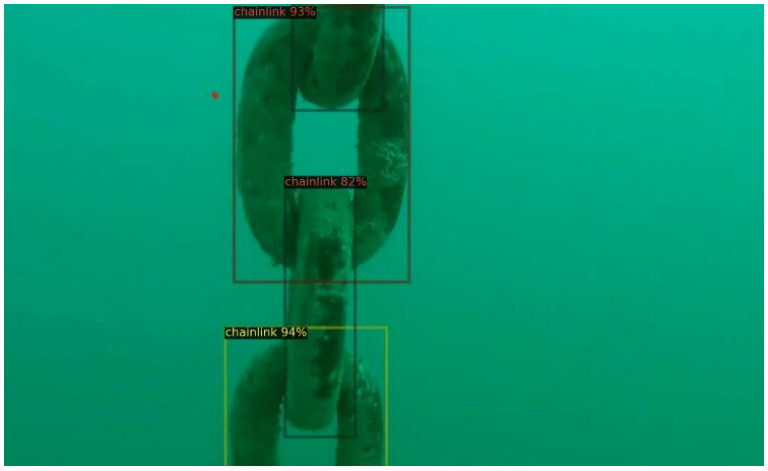


Figure 51. The result of identification confidence threshold being not properly set.

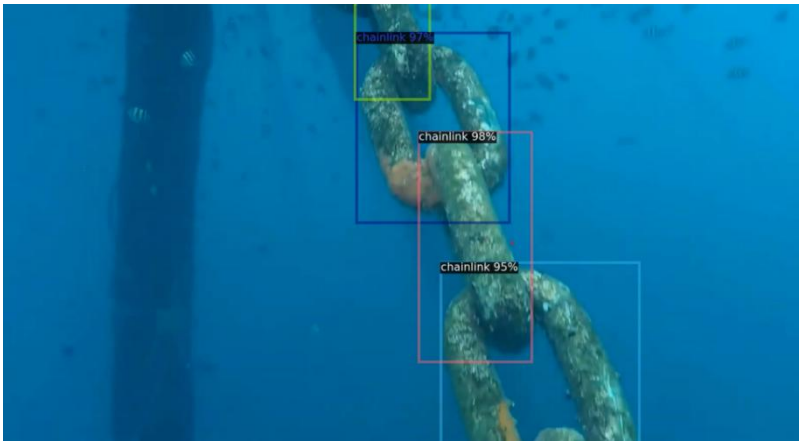
In this example, numerous false positives with low confidence values are observed, suggesting limitations in the model's precision. To enhance identification accuracy, the model underwent retraining and refinement. Subsequently, the identification model was tested on an underwater operational scenario video, with representative results presented in Figure 52.

(a) Identification result 1



(b) Identification result 2

(c) Identification result 3



(d) Identification result 4

Figure 52. The object identification results.

Each bounding box corresponds to a detected chain link, with confidence scores indicating the model's certainty regarding the classification. Higher scores reflect increased confidence that the detected object matches the predicted category. In the presented results, the majority of chain links achieve confidence scores above 0.8, with many exceeding 0.9, demonstrating the model's high precision.

The identification program is compatible with both Google Colab and local Python environments. Google Colab offers access to reliable GPU resources, although prolonged usage beyond several hours per day may incur additional fees. Users with Colab Pro membership benefit from enhanced computational power, resulting in faster and more reliable processing. The computational specifications available through Colab Pro are summarized in Figure 53.

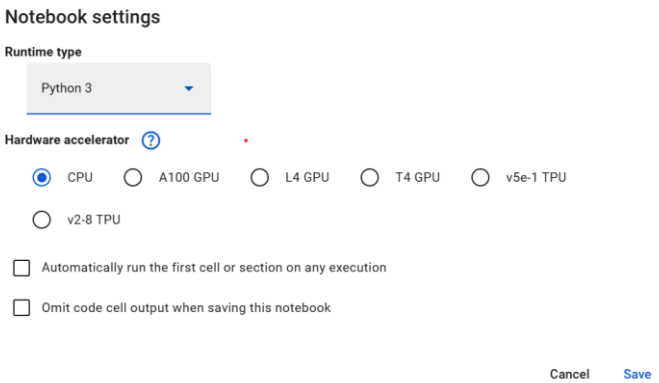


Figure 53. The computational resources provided for Colab Pro.

Google Colab typically provides access to the NVIDIA T4 GPU, which offers solid performance suitable for many tasks. This resource is particularly accessible to users without paid subscriptions, although it is subject to usage time limitations. Initial sessions may allow up to 10 hours of continuous use; however, with frequent usage, the allotted runtime gradually decreases.

The identification model can also be downloaded from a cloud repository for local deployment, enabling users to perform identification tasks without retraining the model. This approach is advantageous when local training is impractical or when computational resources are insufficient to achieve high-precision models. The source for downloading the identification model is illustrated in Figure 54.

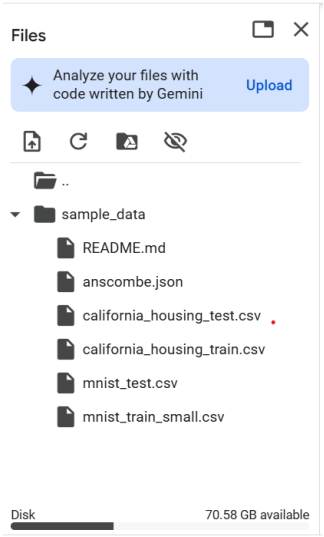


Figure 54. The place that the identification model can be downloaded

Running the identification program locally using Python is free; however, performance is highly dependent on the hardware configuration. Limitations in GPU memory can lead to slower training. Moreover, training quality may be adversely affected when using older or less powerful GPUs, even if training parameters remain consistent. For example, Figure 55 presents an identification model trained on an NVIDIA GeForce RTX 2050 GPU.

Figure 55. The identification model trained by NVIDIA GeForce RTX 2050

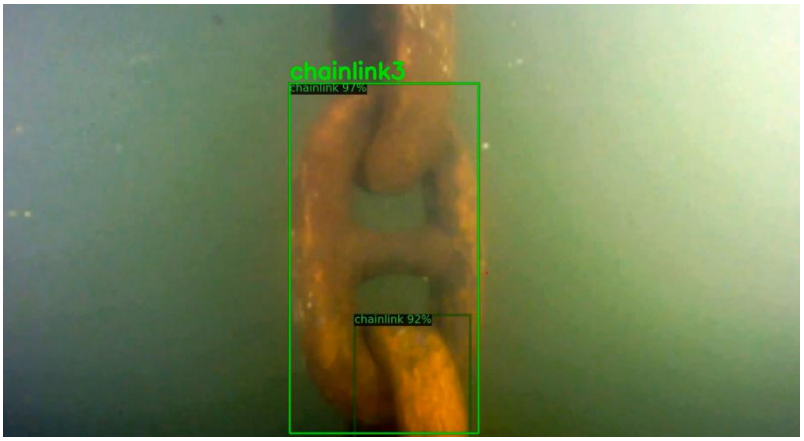
Comparing Figure 55 with Figure 52 (a), it is evident that Figure 55 shows lower confidence levels, indicating reduced precision.

3.2.2. Object labelling

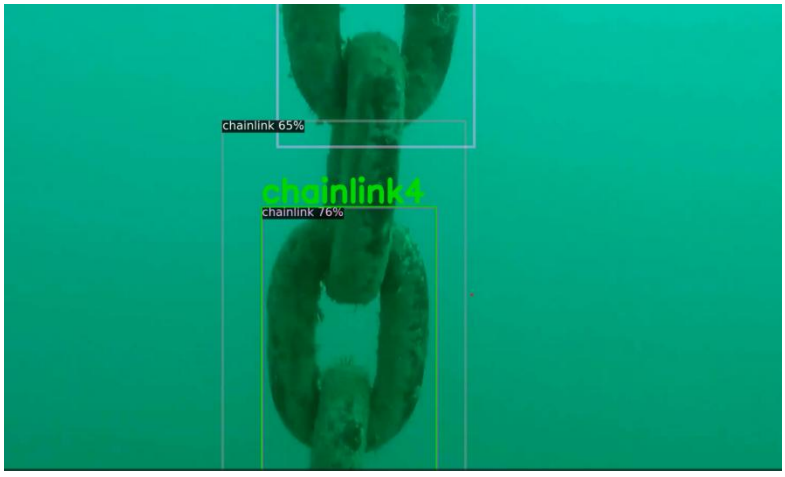
Object labelling is a crucial step in the identification process that enables accurate localization of targets. In the context of underwater operations, labelling chain links within a mooring chain is essential for aligning and locating specific links, thereby supporting efficient inspection and maintenance activities.

The object labelling workflow closely resembles that of object identification. However, a key distinction lies in the application of identification false-prevention methods during the testing phase. These methods help preserve accurate detections and eliminate false positives, particularly when the model sequentially numbers the detected bounding boxes. This ensures a more consistent and reliable labelling outcome.

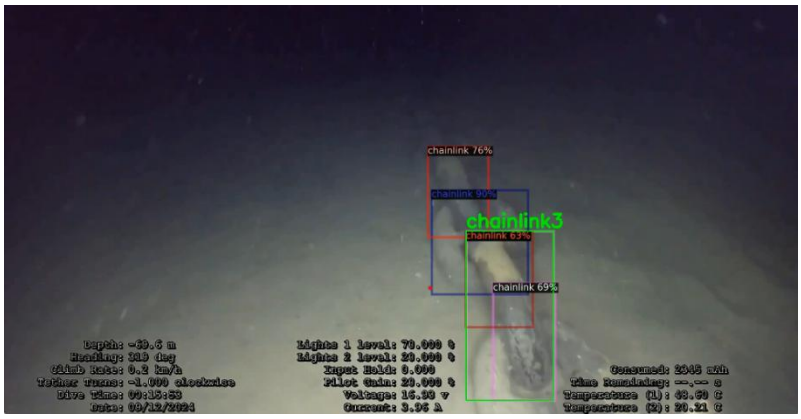
To assess the feasibility of this approach, an object labelling test was conducted using a video captured in an underwater operational scenario. The corresponding results are presented in Figure 56.



(a) Numbering result 1



(b) Numbering result 2



(c) Numbering result 3

(d) Numbering result 4

Figure 56. The chain labelling result.

As illustrated in Figure 56, the labelling program exhibits a reasonable degree of robustness and accuracy. However, its performance remains highly sensitive to various operational conditions. Critical factors influencing the effectiveness of the system include the velocity of the underwater robot, the resolution and quality of the imaging equipment, and the number of chain links visible within each video frame. To enhance labelling precision and consistency, it is advisable to operate the robot at reduced speeds, utilize high-quality cameras, and limit the number of chain links captured per frame.

3.3. Underwater localization

Underwater localization follows the identification phase and provides crucial spatial information about detected objects relative to the camera. This step supports operational decision-making by guiding movement and positioning. There are two types of methods introduced in this section. Method 1 is based on deep learning and Method 2 is based on trigonometry.

3.3.1. Method 1

Method 1 is built on the deep learning algorithm Metric 3D, a high-performance and reliable depth estimation model for monocular cameras. Metric 3D ranks high on the routing KITTI and NYU benchmarks. The performance comparison between Metric

3D and other algorithms is indicated in Table 5.

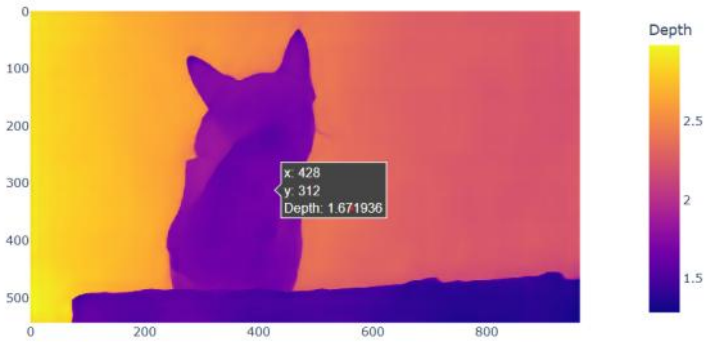
Table 5. The performance indication of Metric 3D algorithm.

Algorithm	Backbone	KITTI δ1 ↑	KITTI δ2↑	KITTI AbsRel ↓	KITTI RMSE ↓
ZoeDepth	ViT-Large	0.971	0.995	0.053	2.281
ZeroDepth	ResNet-18	0.968	0.996	0.057	2.087
IEBins	SwinT-Large	0.978	0.998	0.050	2.011
DepthAnything	ViT-Large	0.982	0.998	0.046	1.985
Metric 3D	ViT-Large	0.985	0.998	0.044	1.985
Metric 3D	ViT-giant2	0.989	0.998	0.039	1.766

The δ represents accuracy and the AbsRel and RMSE represent errors. From Table 10, it could be found that the backbone of Metric 3D has the highest accuracy and lowest errors among all metric depth estimation, proving its good performance on depth estimation.

Metric 3D takes a single RGB image as input and produces a depth map, where each pixel's value indicates its distance from the camera (in meters). The model is pretrained, allowing immediate use without the need for additional training. If an image is passed through the model, it outputs a complete depth map corresponding to the visual structure of the input. An example is shown in Figure 57.

(a) Raw image



(b) Predicted depth map

Figure 57. Raw image and depth map acquired after the processing of Metric 3D.

As illustrated, the generated depth map effectively captures the depth structure present in the raw image, demonstrating the model's capability to translate visual input into spatial information. To better explain the underlying approach, the procedural steps of Method 1 are outlined below:

Depth map prediction

Using the Metric3D algorithm, depth estimation is performed with the help of pre-trained models. The available models include the ConvNeXt-L series and the Vision Transformer (ViT) series. In this assignment, a ViT-based model is selected due to its clearer implementation guidance and ease of integration. The ViT series includes three model variants: metric3d_vit_small, metric3d_vit_large, and metric3d_vit_giant2. Generally, model size is positively correlated with prediction accuracy but also leads to increased computational demand. The memory footprint and expected accuracy of each ViT model are summarized in Table 6.

Table 6. The occupied space and the precision of models in ViT.

Name	Size	Precision
Metric3d_vit_small	143M	Low
Metric3d_vit_large	1.5G	Medium
Metric3d_vit_giant2	5.51G	High

In this assignment, the metric3d_vit_small model is selected for depth estimation, as it provides sufficient accuracy while maintaining a manageable computational load.

Compared to deep learning models used for underwater object identification, the Metric3D algorithm is significantly more computationally intensive. Depth map prediction typically requires more processing time than object identification, particularly when using larger models. As such, it is recommended to perform depth estimation on systems equipped with high-performance GPUs. Alternatively, cloud-based platforms such as Google Colab can be utilized, offering access to powerful hardware that can support the demands of this algorithm.

Bounding box acquisition

For underwater localization using the Metric 3D method, the identification model must first be obtained, as outlined in Section 3.2. The input video is then divided into individual frames to facilitate object detection. These frames are processed by the identification model to generate bounding boxes around detected objects. The resulting bounding boxes are subsequently used in conjunction with the depth maps to estimate the distances of the identified objects from the camera.

Distance acquisition

Once both the depth map and bounding boxes are obtained, depth (distance) information for the entire image and the localized objects becomes available. Conceptually, when the bounding boxes are overlaid onto the depth map, each enclosed region contains numerous pixels, each representing a specific depth value. However,

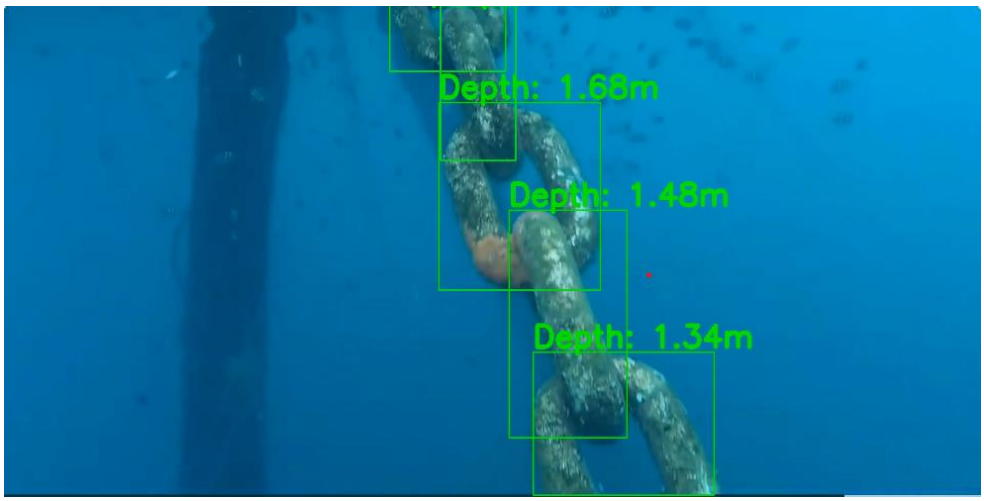
the distance of the object cannot be directly determined from this data; further processing of the depth values within each bounding box is required. Common processing approaches include minimum distance extraction, average distance calculation, and maximum distance extraction.

The choice of method depends on the operational scenario. Minimum distance extraction is used when it is important for the camera to maintain a safe distance from the object or when the object contains hollow sections, which may distort average values. Average distance calculation is appropriate when a general estimation of the object's distance from the camera is sufficient. Maximum distance extraction is applied in scenarios where the object has no hollow sections and the camera is intended to move closer to the object—although this is relatively uncommon.

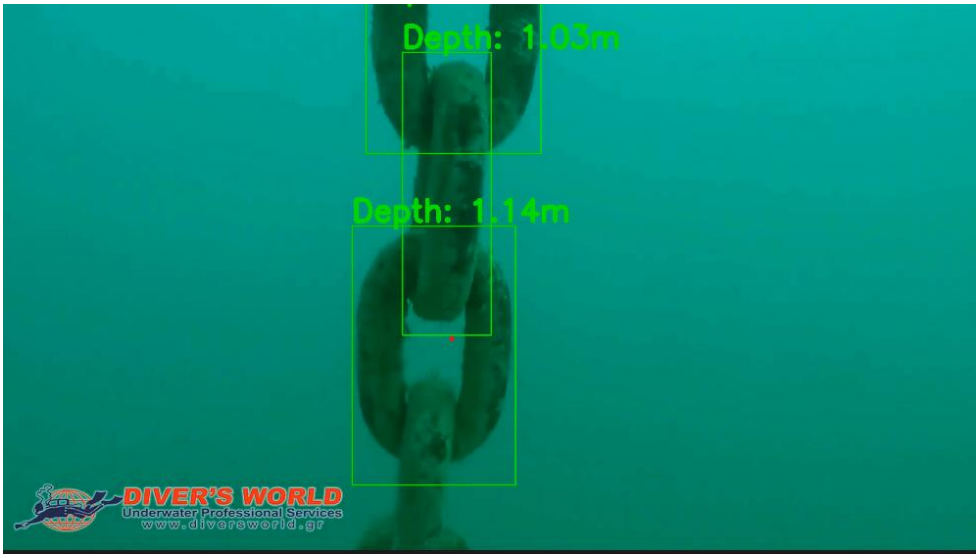
In underwater robotic operations, such as the inspection of mooring chains, the chain links typically feature hollow sections, and it is desirable to avoid close proximity between the robot and the object. Therefore, the minimum distance extraction method is applied. Once the distance is calculated, it is annotated directly onto the processed images for further analysis or visualization.

Video formation

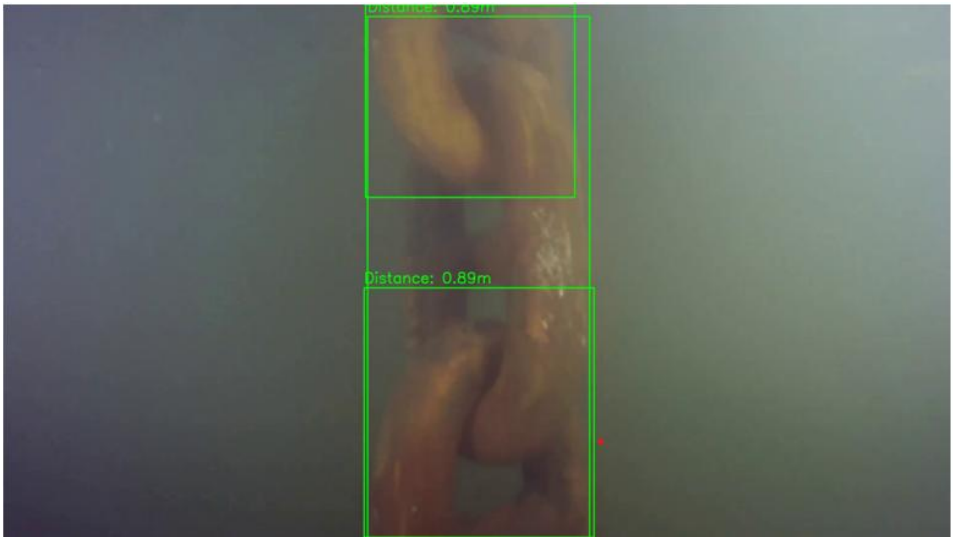
Ultimately, the processed images containing distance information are recombined to form a complete video. This final video includes both bounding boxes and corresponding depth annotations for each detected object. Prior to the final deployment, preliminary localization results using Method 1 are presented in Figure 58.



(a) Localization scenario 1



(b) Localization scenario 2



(c) Localization scenario 3

Figure 58. The localization results of Method 1.

In scenario (c), the distance between the chain link and the camera is approximately one meter. The predicted distance of 0.89 meters closely aligns with this actual value, demonstrating the model's accuracy. In scenario (a), three depth values are indicated; as expected, chain links that appear farther from the camera are assigned higher distance values. This further confirms the method's reliability. Visually, the chain link in scenario (b) is estimated to be about one meter from the camera, and the model's output supports this observation. These examples suggest that Method 1 achieves sufficient accuracy for practical underwater distance estimation.

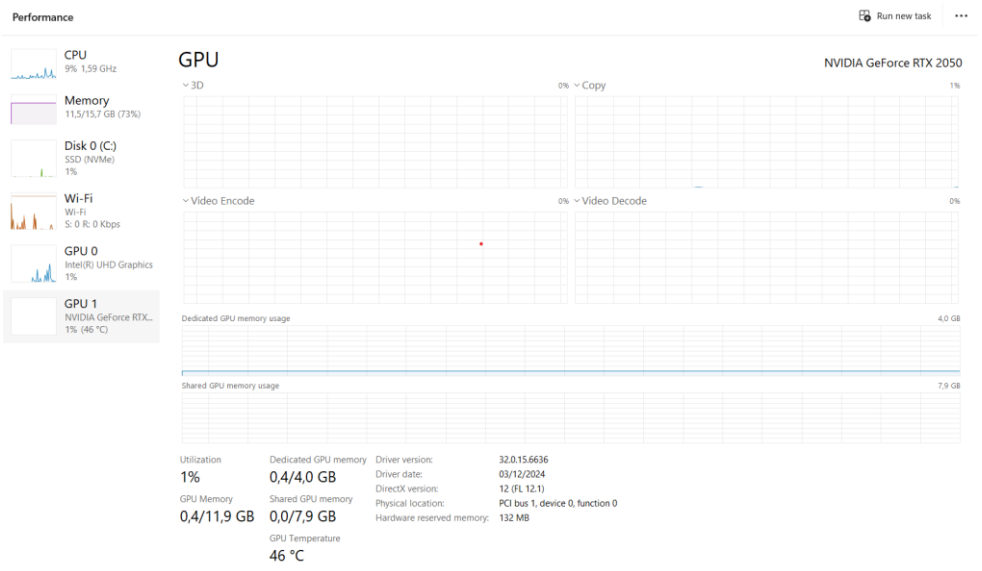
A key advantage of Method 1 is that it does not require prior knowledge of the object's dimensions. This makes it particularly suitable for underwater scenarios where the object size is unknown or variable. Unlike depth cameras, which are effective but costly and not widely used underwater, Method 1 requires only a monocular camera, significantly reducing hardware constraints. For reference, one example of a

commercial depth camera is shown in Figure 59.

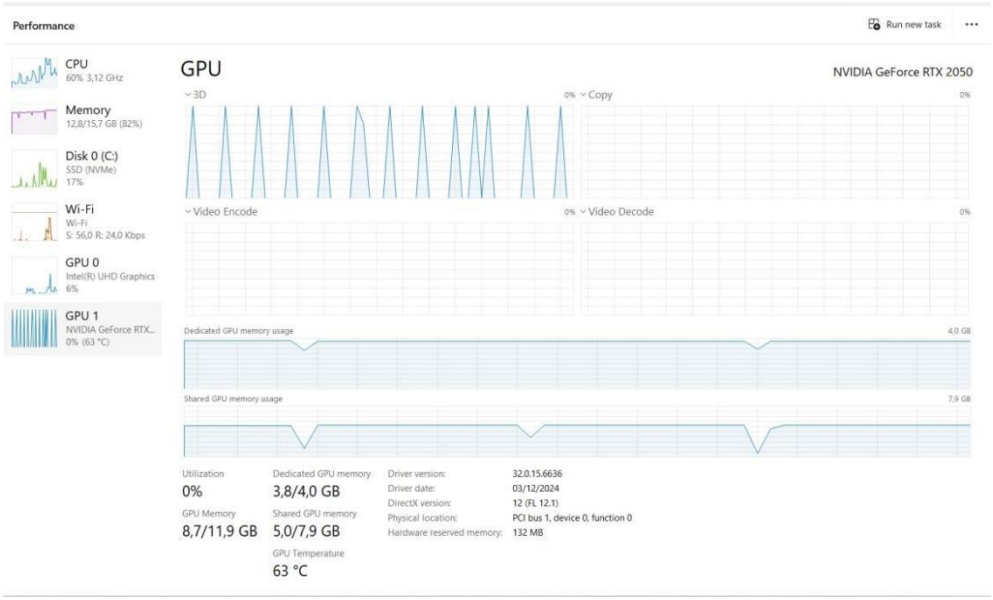
Figure 59. Intel RealSense Depth Camera D435 [145]. The price of the camera is around 400 euros

In contrast to such hardware, Method 1 enables depth estimation without specialized equipment, thereby broadening its applicability to a wide range of underwater operations.

However, the method is not without drawbacks. A significant limitation is its high computational demand. Since underwater object identification already involves deep learning algorithms, adding a second deep learning model for localization increases the computational burden—particularly on the GPU. In many cases, this results in memory overload and runtime errors such as “CUDA out of memory” in Python. These issues typically arise when GPU resources are insufficient to handle the combined workload of identification and localization tasks. Greater attention should therefore be given to optimizing GPU memory usage during processing. The state of the laptop before and during localization using Method 1, as well as a common CUDA error message, are illustrated in Figure 60.



(a) Laptop state before localization



(b) Laptop state during application of Method 1

```
     50 frame_tensor = torch.from_numpy(frame_rgb).unsqueeze(0).to(device)
     52 with torch.no_grad():
---> 53     pred_depth, _, _ = model.inference({'input': frame_tensor})
     55 pred_depth_np = pred_depth.squeeze().cpu().numpy()
     56 outputs = predictor(frame)

File ~/.cache\torch\hub\yvanyin_metric3d_main\mono\model\monodepth_model.py:12, in DepthModel.inference(self, data)
     10 def inference(self, data):
     11     with torch.no_grad():
---> 12         pred_depth, confidence, output_dict = self.forward(data)
     13     return pred_depth, confidence, output_dict

File ~/.cache\torch\hub\yvanyin_metric3d_main\mono\model\model_pipelines\__base_model__.py:13, in BaseDepthModel.forward(self, data)
     12 def forward(self, data):
---> 13     output = self.depth_model(**data)
     15     return output['prediction'], output['confidence'], output

File c:\Users\enron\AppData\Local\Programs\Python\Python312\Lib\site-packages\torch\nn\modules\module.py:1511, in Module._wrapped_call_impl(self, *arg
   1509     return self._compiled_call_impl(*args, **kwargs)  # type: ignore[misc]
   1510 else:
-> 1511     return self._call_impl(*args, **kwargs)
...
--> 445 attn = attn.softmax(dim=-1)
    446 attn = self.attn_drop(attn)
    448 x = (attn @ v).transpose(1, 2).reshape(B, N, C)

OutOfMemoryError: CUDA out of memory. Tried to allocate 2.59 GiB. GPU 0 has a total capacity of 4.00 GiB of which 0 bytes is free. Of the allocated me
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

(c) CUDA out of memory error

Figure 60. The laptop state illustration and the possible CUDA error regarding Method 1.

3.3.2. Method 2

Although Method 1 offers a generally applicable approach to localization, it poses significant demands on computational hardware, particularly the GPU. To address this limitation, Method 2 has been developed as a lightweight alternative that requires less computational power.

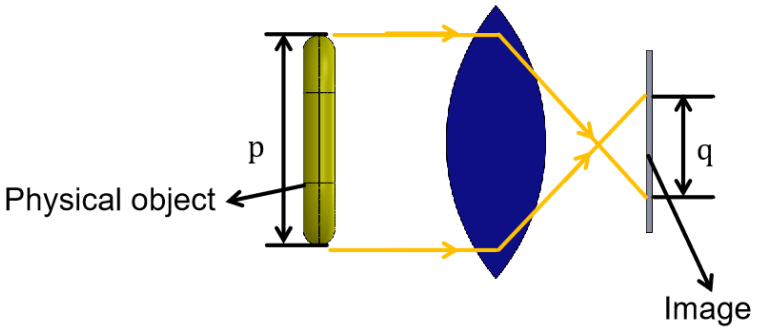
Method 2 is grounded in basic principles of trigonometry and leverages the known physical dimensions of the object to estimate the distance and visual angle between the camera and the object. This method relies on camera imaging principles and geometric similarity, providing a transparent, white-box approach as opposed to the “black-box” nature of deep learning in Method 1. The formulas used to calculate the distance d and viewing angle α are shown below:

$$\frac{\mathrm{d}}{f} = \frac{p}{q} \tag{12}$$

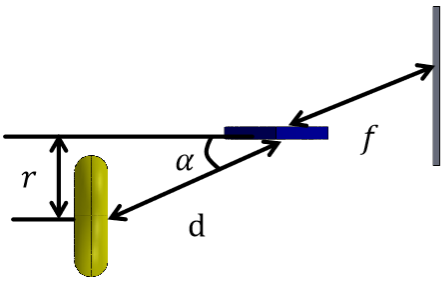
$$\mathrm{d} = \frac{\mathrm{fp}}{q} \tag{13}$$

$$\alpha = \sin^{-1}\frac{\mathrm{r}}{\mathrm{d}} \tag{14}$$

Where d is the distance between the camera and the object, f is the focal length of the camera, p is the height of chain, which is obtained before the detection, q is the height of chain link in the sensor, and r is the chain deviation from central line. Height of chain in the sensor p and chain deviation from central line r are obtained during the detection. The conceptual schematic of Method 2 is presented in Figure 61.



(a) Main view



(b) Top view

Figure 61. The principle of Method 2

To better explain the principle of Method 2. The working procedures are indicated as follows:

Camera parameter & object size acquisition

Before processing, the camera's specifications—focal length and pixel size—must be obtained, typically from the camera's datasheet. In addition, the real-world dimensions of the object are required. For this study, the dimensions of the chain link model (width, height, length) are known. Since the height of the chain link remains consistent during robotic movement while the width may vary during detection, the height is selected as the value of p in equations (12) and (13).

Bounding box acquisition

Please refer to ‘Bounding box acquisition’ in 3.3.1.

Real-time parameter acquisition

Thanks to the object identification process, the height of the detected bounding box is used as the estimated height of the chain link in the image, denoted as q in Equations (12) and (13). Additionally, the horizontal deviation of the object from the image centre, represented as r in Equation (14), is calculated by measuring the distance between the centre point of the bounding box and the vertical central axis of the image.

Distance calculation

The distance d between the object and camera can be calculated using equation (13).

Video formation

As with the previous methods, the individual images obtained during the ‘Bounding Box Acquisition’ step are reassembled into a complete video. The resulting video should display, at a minimum, both the distance and angle between the camera and the detected object.

Before conducting the full experiment, Method 2 was initially applied to an underwater operational scenario video featuring a mooring chain. While most detections yielded accurate distance estimates, a few incorrect results were observed, as illustrated in Figure 62.

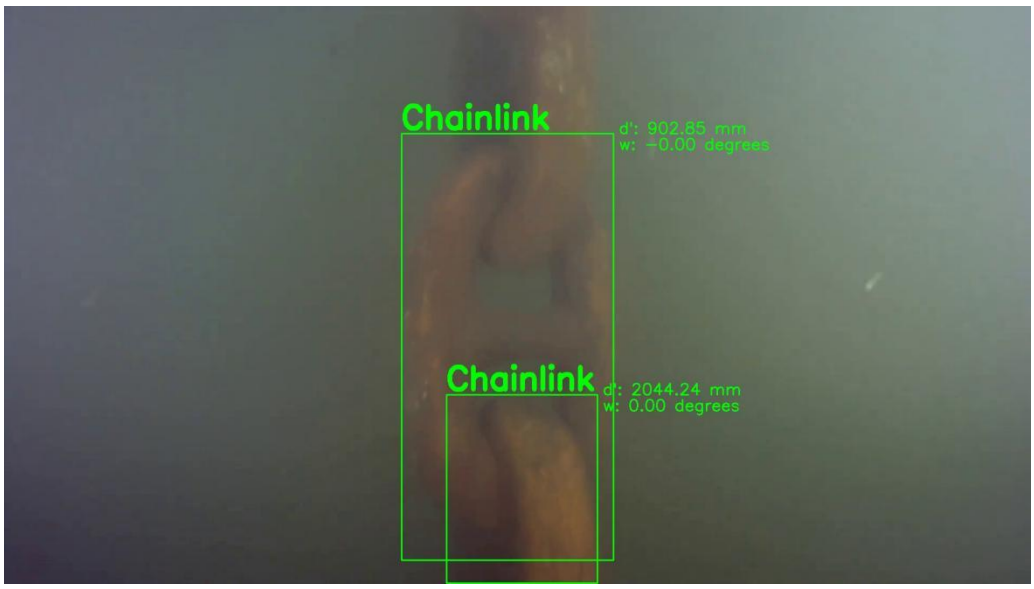
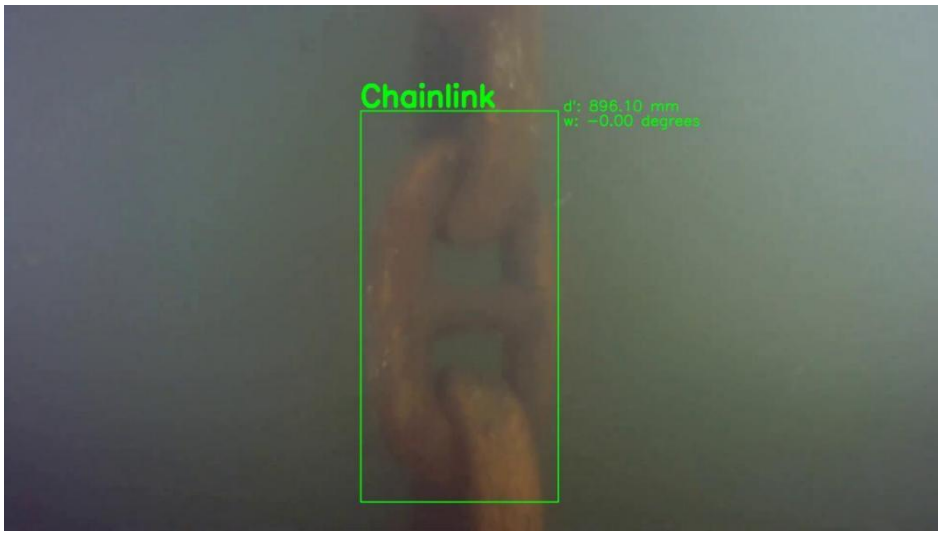


Figure 62. Some wrong detections in Method 2.

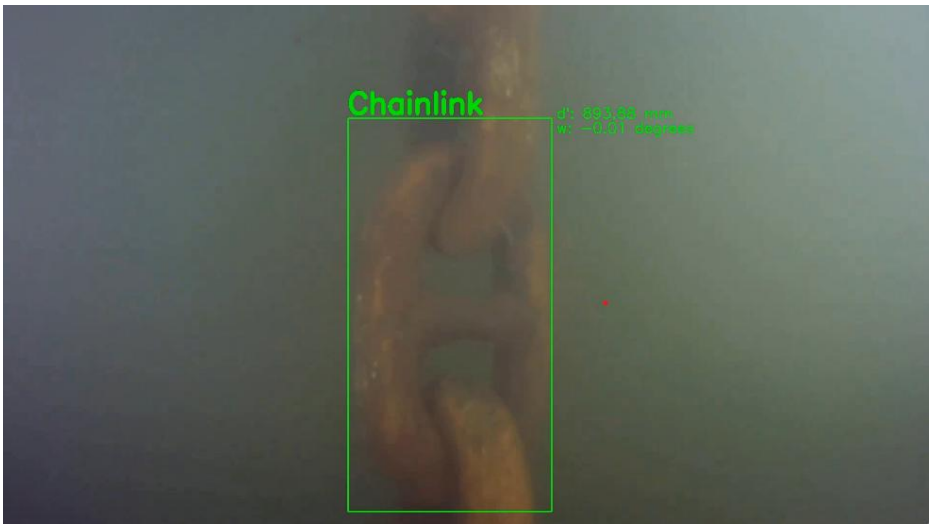
Incorrect distance estimations were primarily caused by incomplete chain links appearing in the image frame. In such cases, the height of the bounding box q was halved, resulting in an overestimated distance d, which could be nearly twice the actual value.

To address this issue, an edge filtering technique was introduced. This approach ensures that none of the four edges of the bounding box touch the boundaries of the image. Any bounding box that fails this condition is excluded from the calculation. The

improved results, after applying edge filtering, are shown in Figure 63. The parameters of the camera are indicated in Table 7.



(a) Result 1



(b) Result 2

Figure 63. The results that apply edge filtering in Method 2.

Table 7. The camera parameter of Low-Light HD USB Camera [12].

Camera parameter	**Value**
Focal length	2.97 mm
Pixel size	2.8μm (H) x 2.8μm (V)

From the localization results in Figure 63, it is evident that the estimated distance is approximately 0.9 meters, closely aligning with the actual distance. This demonstrates that Method 2 can achieve a high level of accuracy under appropriate conditions. However, applying Method 2 to other videos can be more challenging than Method 1, particularly when camera-specific parameters such as pixel size and focal length are unavailable. The relationship between focal length and pixel size can be written as

follows:

$$\mathrm{Px} \cdot \mathrm{H} = \frac{dq}{f} \tag{15}$$

Where H is the height of the chain link in pixel on the camera, Px is the pixel size of the camera.

If the Px and f are wrongly set in a scenario, the distance result will be wrong, a false distance indication using Method 1 is shown in Figure 64.

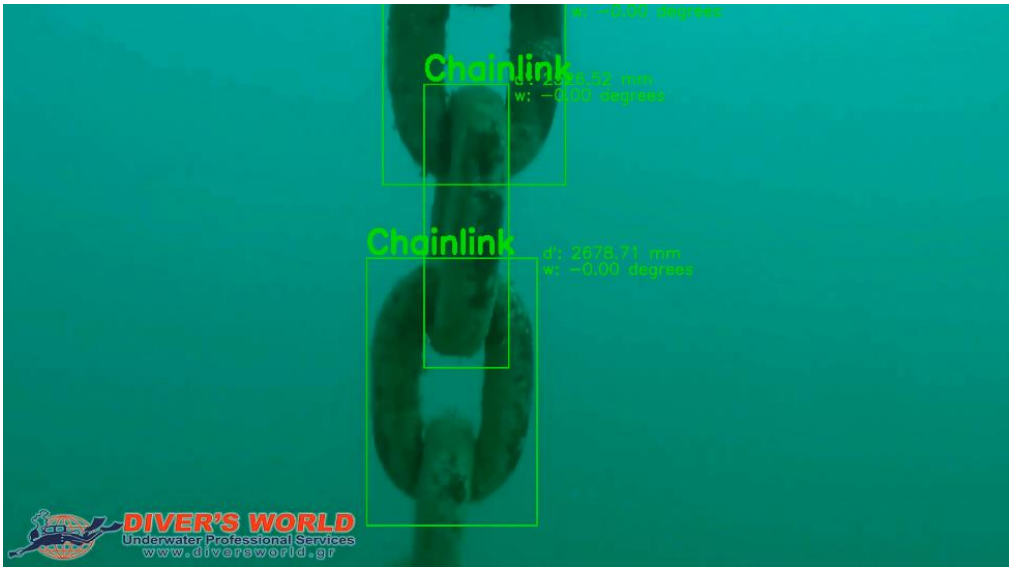


Figure 64. A false distance indication using Method 2.

Despite this limitation, Method 2 offers several advantages. First, it significantly reduces the computational burden, leading to improved processing efficiency. Second, it provides richer output, including both the distance and visual angle between the object and the camera, which can enhance underwater operational control. Additionally, unlike black-box deep learning models, Method 2 relies on clearly defined geometric principles, making the process more interpretable and its results more persuasive. The laptop's performance during the application of Method 2 is illustrated in Figure 65.

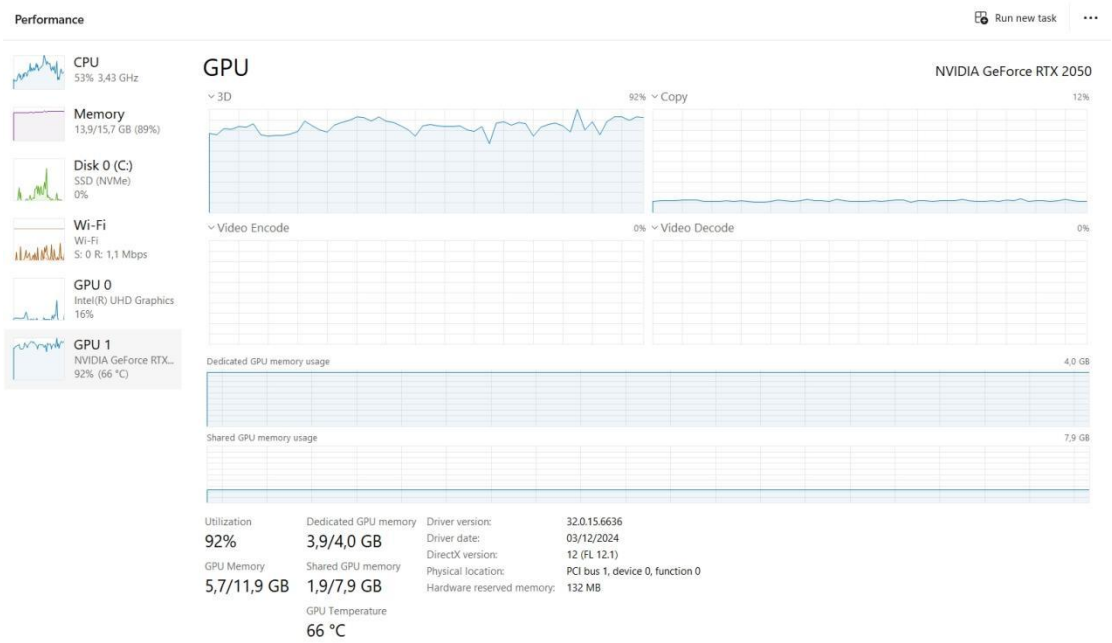


Figure 65. The laptop state during the application of Method 2.

Compared to Figure 60 (b), both the occupied GPU memory and shared GPU memory usage in Method 2 are significantly lower than those observed in Method 1. This clearly demonstrates that Method 2 imposes a lighter computational load, making it more suitable for systems with limited processing capabilities.

However, Method 2 does come with certain limitations. Specifically, it requires prior knowledge of the object's dimensions and camera parameters. While camera specifications—such as focal length and pixel size—are generally available from datasheets, obtaining accurate object dimensions underwater can be challenging. This is particularly true for irregular or unknown targets such as shipwrecks, marine organisms, or nonstandard underwater structures, where predefined measurements are not readily accessible. This presents a key obstacle in applying Method 2 to diverse underwater environments. Another limitation arises from the reliance on bounding box height for distance calculation. The accuracy of the distance estimate is highly sensitive to the precision of the bounding box. Errors in object detection may lead to inaccurate bounding box dimensions, particularly the height, which directly impacts the calculated distance. This issue is illustrated in Figure 66.

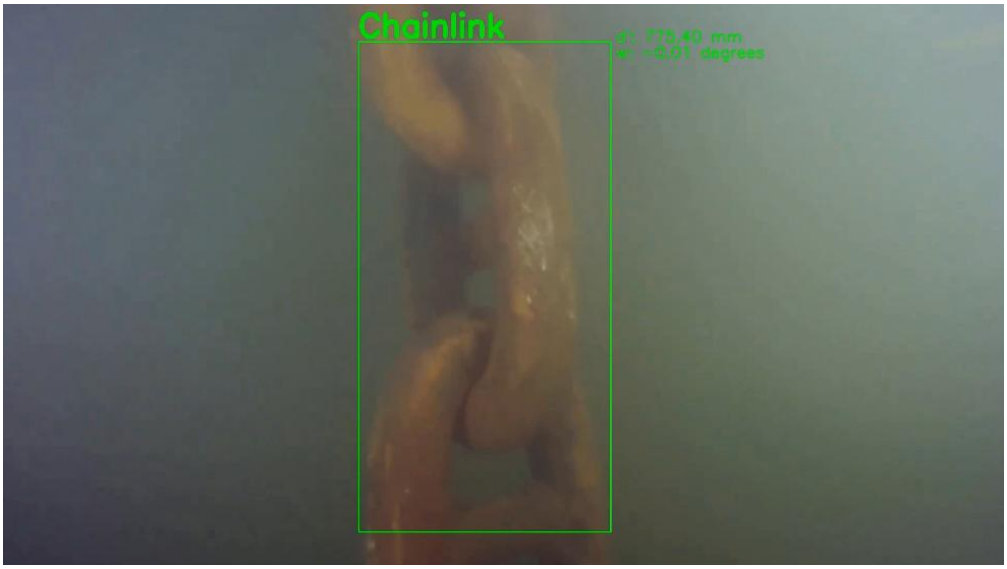


Figure 66. The error of bounding box height.

In this scenario, the bounding box height is overestimated, resulting in a calculated distance that is smaller than the actual distance. This type of error is more pronounced in Method 2 than in Method 1. In Method 1, distance estimation is based on the depth values of individual pixels within the bounding box, which are typically spatially consistent and less sensitive to the bounding box's exact size. In contrast, Method 2 relies on a geometric relationship in which the estimated distance is inversely proportional to the bounding box height. As a result, any overestimation in bounding box height leads directly to an underestimation of distance. Despite this limitation, such errors do not occur frequently, and their overall impact on the accuracy of the method is limited. Thus, the error can be considered acceptable for most practical applications.

Given these characteristics, Method 2 is best suited for scenarios where the object size is known and computational resources are limited, such as when using standard laptops without high-performance GPUs. Additionally, due to its low computational demand, Method 2 is more suitable for real-time processing compared to Method 1.

4. Experiments

To verify the feasibility of the real-time identification and localization method proposed in the report. The experiment is carried out. It evaluates object identification and localization methods in a realistic underwater environment. Video footage captured by the ROV (Remotely Operated Vehicle) is uploaded to Google Colab in the form of screenshots for processing. This experiment effectively simulates underwater detection thanks to the fact that the detection is carried out underwater.

4.1. Experiment setup

The experiment is designed to achieve simultaneous identification and localization of a chain link in an underwater environment. Based on the distance data provided through localization, the TPU grabber mounted on the ROV is expected to accurately clamp the chain link under the guidance of the pilot. To facilitate this setup, the materials required for the experiment are listed in Table 8.

Table 8. The required material list in the experiment

Name	Quantity	Description
Chain link	1	3D-printed using PETG to simulate a real chain link
Weight	1	Attached to the chain link and placed at the bottom to prevent floating
TPU grabber	1	A flexible 3D-printed gripper (TPU) mounted on the ROV to clamp the chain link
Connector	1	A PLA 3D-printed component that connects the TPU grabber to the ROV
ROV	1	Serves as the underwater mobile platform for video capture and manipulation
Laptop	1	Used for real-time data processing and to guide the pilot's operation

With the materials at hand, the experiment setup is arranged as shown in Figure 67:

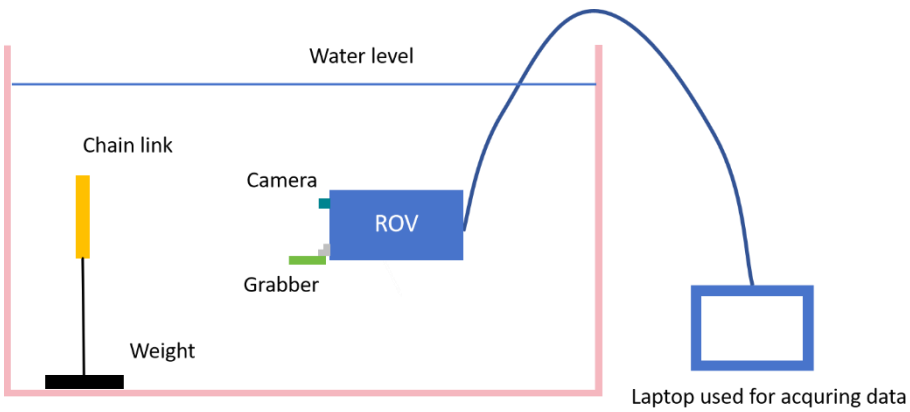


Figure 67. The experiment setup.

4.2. Experiment procedure

It is assumed that the experiment is set up as illustrated in Figure 68. Firstly, to accurately measure the actual distance between the chain link and the ROV, a ruler is affixed to the side of the experimental pool. This allows for easy distance measurements during operation. The ruler used in the experiment is shown in Figure 68.

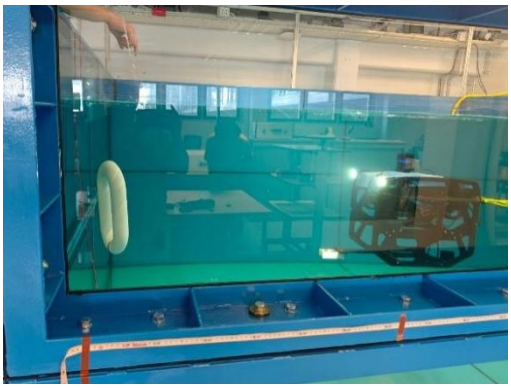
Figure 68. The applied ruler in the experiment. The chain link is placed at the position which corresponds to 0 in the ruler

Secondly, the pilot controls the ROV toward the chain link. A second person is responsible for recording the time, the actual measured distance (based on the ruler), and the estimated distance provided by the identification and localization system.

Finally, Once the ROV approaches the target, the pilot uses the distance data output by the program to guide the TPU grabber and attempt to clamp the chain link. The entire procedure is repeated multiple times to enhance the accuracy and reliability of the results.

5. Experimental results

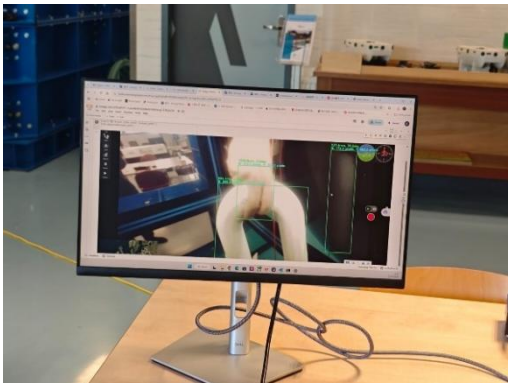
Throughout the experiment, the pilot first controls the ROV to approach the chain link gradually. Then, the ROV returns to its original place. This process is repeated several times to collect enough experimental data. Figure 69 shows some important scenes during the experiment.



(a) Approaching the chain link

(b) Clamping the chain link

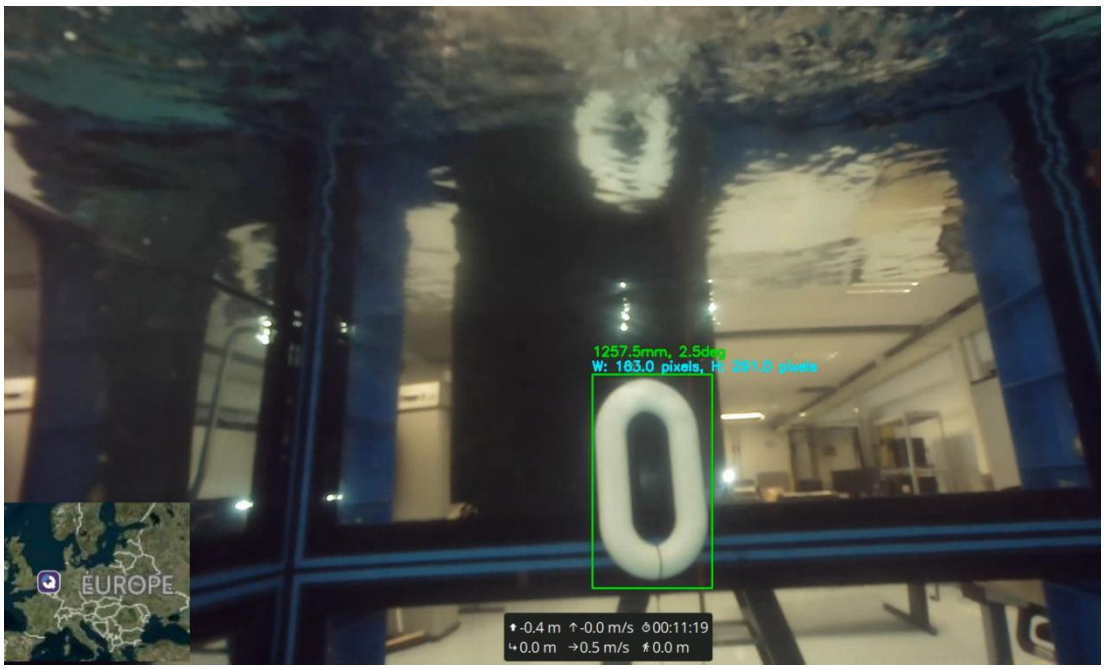
(c) The application that receives video data



(d) Online platform which process the images

Figure 69. Some important scenes during the experiment.

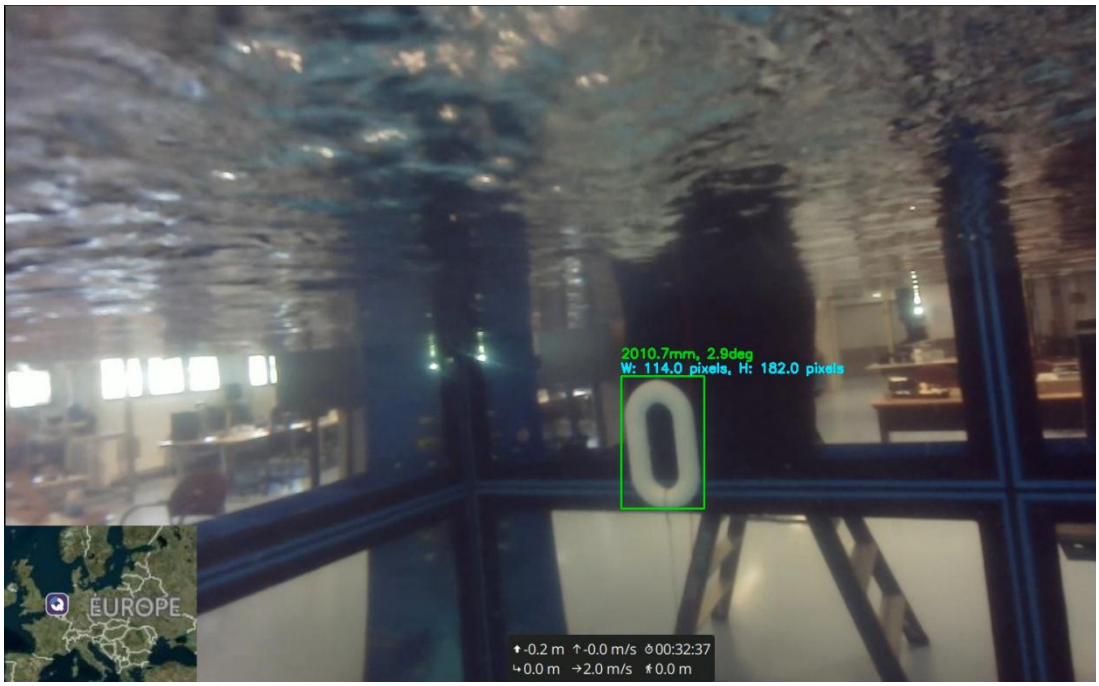
During the approaching of the ROV, some visual results of the identification and localization of the chain link are indicated in Figure 70.



(a) Experiment result 1

(c) Experiment result 2

(d) Experiment result 3



(e) Experiment result 4

Figure 70. Visual results of the experiment.

The results demonstrate that the program is capable of accurately identifying and localizing the chain link, particularly when it is in close proximity to the ROV's camera. However, there are occasional instances where the chain link is not successfully detected, indicating room for improvement in the object identification model.

To assess the accuracy of the localization, a comparison between the actual measured distances and the estimated distances provided by the program is summarized in Table 9.

Table 9. The comparison between actual measured distance and distance obtained by the localization program in the experiment.

Order	Actual measured distance (unit: m)	Obtained distance from the program (unit: m)	Error
1	1.0	1.2	20.0%
2	1.3	1.5	15.4%
3	1.0	1.3	30%
4	0.6	0.62	3.3%
5	1.5	1.2	20.0%
6	0.7	0.65	7.1%
7	2.0	2.0	0
8	1.45	1.5	3.4%
9	2.0	1.6	20.0%
10	0.9	1	11.1%

As shown in the table, the localization errors are generally within 30%, which may

be due to the imperfect bounding box and the delay of online processing. The result suggests that the method delivers acceptable accuracy for practical underwater operations. This level of precision is likely sufficient to support operator decision-making during ROV manipulation tasks.

It is worth noting that only Method 2 was used in this experiment, as it has been observed to produce localization results comparable to Method 1, while offering advantages in computational efficiency and real-time performance.

6. Conclusions and recommendations

6.1. Conclusions

This study investigated an integrated system for underwater object identification and localization using ROV-acquired video data, aiming to detect and locate chain links for robotic grasping via a TPU grabber.

The object identification component is based on a deep learning detection model trained on labelled chain link images captured in underwater environments. The model processes each video frame to identify and generate bounding boxes around chain links in real time. These bounding boxes serve as critical inputs for both localization methods. The accuracy and consistency of detection directly impact localization performance, particularly for methods that rely on bounding box geometry. While the model performs reliably in most conditions, occasional misdetections occur due to visual disturbances such as turbidity, reflections, or partial occlusion. Nonetheless, the identification model demonstrates strong performance when the chain link is clearly visible, offering a robust foundation for subsequent localization.

Method 1 estimates distance by computing the average depth of all pixels within the detected bounding box using pixel-wise depth estimation. This method requires no prior knowledge of the object's dimensions or the camera's intrinsic parameters, making it adaptable to a wide variety of underwater objects, including unknown structures or marine life. However, its reliance on dense depth estimation results in high computational demands, which may limit its real-time applicability on devices with limited processing power. Despite this, Method 1 is especially useful in scenarios with inconsistent bounding box results or when object dimensions are unavailable.

Method 2 calculates distance by leveraging the height of the bounding box in pixels, the known physical dimensions of the object, and the intrinsic parameters of the camera, such as focal length and pixel size. This geometric approach offers a lightweight, efficient alternative to pixel-based depth estimation, making it well-suited for real-time operation on standard laptops. Method 2 assumes accurate object detection and consistent object size, and while it is more sensitive to errors in bounding box height, these errors occur infrequently and typically remain within acceptable error margins. The method performs best in structured scenarios where the size of the target object is known beforehand.

Experiments confirmed the practicality of the proposed approach in realistic underwater conditions. As the ROV approached the chain link, the system effectively performed simultaneous object detection and localization in real time. Visual results showed consistent identification of the target object, especially when it was closer to the camera. The TPU grabber, guided by the distance output from the localization module, was able to successfully clamp the chain link under pilot control.

The recorded data revealed that the estimated distances deviated from the actual measured values by no more than 30%, which is within an acceptable range for many underwater operational tasks. The accuracy was generally higher when the object appeared centrally and fully within the camera frame, and slight deviations mainly

occurred due to imperfect bounding box detection or Internet dalay. These outcomes demonstrate that the system is capable of providing reliable distance feedback to support ROV operation in structured underwater environments.

Overall, the proposed method meets the essential requirements for real-time object localization and is particularly effective when both the object's physical dimensions and camera parameters are known in advance. This makes it a promising solution for underwater tasks involving object manipulation, such as grasping or inspection.

6.2. Recommendations

Based on the experimental outcomes and performance evaluation, several recommendations are proposed to guide future research and enhance practical deployment:

6.2.1. Improve Detection Accuracy

The object detection component occasionally failed when the chain link was located farther from the camera, partially occluded, or affected by underwater lighting variations. To address this, the detection model could be enhanced by training with a more diverse dataset that includes various object orientations, lighting conditions, and occlusion scenarios. Augmenting the dataset with synthetic underwater images or applying domain adaptation techniques may further improve the model’s robustness in real-world applications.

6.2.2. Refine Bounding Box Filtering

Distance estimation in Method 2 heavily depends on the accuracy of the bounding box height. While the current edge filtering strategy helps eliminate faulty detections, further refinements could involve evaluating bounding box consistency across frames or using geometric constraints (e.g., aspect ratio or symmetry) to reject invalid detections. Incorporating tracking information may also help smooth out frame-to-frame inconsistencies.

6.2.3. Expand to Unknown Object Sizes

Currently, Method 2 requires the physical dimensions of the target object to be known beforehand, limiting its applicability. Future work could explore hybrid approaches that combine geometrical estimation with stereo vision or depth prediction models to infer object size dynamically. This would allow localization of diverse and unstructured underwater targets, such as marine animals, coral formations, or debris, without prior knowledge of their scale.

6.2.4. Integrate a Feedback Loop for ROV Control

Adding a real-time feedback loop between the object detection/localization module and the ROV control system could improve operation accuracy. For example, the ROV could automatically adjust its speed, orientation, or gripper positioning based on distance estimates and detection confidence. Such adaptive control could reduce operator workload and improve grasping performance, particularly in dynamic or uncertain environments.

6.2.5. Optimize Code for Embedded Deployment

Although the method currently relies on GPU-equipped laptops, optimization for embedded systems or edge devices (e.g., NVIDIA Jetson or Coral TPU) would enhance its portability and field usability. Reducing the model size, employing quantization techniques, and streamlining the inference pipeline would make the solution more suitable for long-term autonomous missions in remote or resource-limited underwater settings.

6.2.6. Handle Multiple Object Scenarios

In many practical underwater operations, multiple similar objects may be present within the camera's field of view. Future research should test the system’s ability to identify, track, and distinguish between multiple targets, ensuring that the correct object is localized and manipulated. Enhancements to the object tracking algorithm and integration of object ID continuity across frames would be critical to maintaining target fidelity.

By implementing these recommendations, the system can become more robust, adaptive, and suitable for a wider range of underwater tasks, from infrastructure inspection to marine exploration and environmental monitoring.

Appendix

A. Scientific research paper

Real time visual-based assisted navigation for underwater robotics

Enrong Xiang
ME – M&TT
TU Delft
Delft, Netherlands

Filippo Riccioli
SAOS – M&TT
TU Delft
Delft, Netherlands

Pooria Pahlavan
SAOS – M&TT
TU Delft
Delft, Netherlands

Jovana Jovanova
ME – M&TT
TD Delft
Delft, Netherlands

***Abstract* - Underwater object detection (i.e. identification and localisation) has gained increasing attention due to its applications in research and exploration of marine environments. Existing studies typically address either identification or localisation separately, lacking a unified approach for underwater object detection. This paper explores simultaneous underwater object identification and localisation using one underwater camera. Identification is achieved through an RCNN (Region-based Convolutional Neural Network), while two approaches are used for localisation: the Metric3D depth estimation algorithm, and an algorithm based on camera imaging principles. The RCNN delivers reliable identification results, and the localisation methods provide distance estimates with accuracy higher than that of the YOLO series, another popular identification algorithm. Experiments conducted in an underwater environment demonstrated the feasibility of the proposed integrated approach. This research contributes to real-time underwater object detection, enhancing underwater vehicle operations for safer and more efficient missions.**

Keywords—Underwater robotics, Identification, Localization, Deep learning, Camera

I. INTRODUCTION

Underwater tasks, such as marine research and underwater exploration, are increasingly popular [1]. However, relying on human divers to perform these tasks is inherently dangerous, expensive, and time-consuming. To address these limitations, the use of underwater robotics, such as ROV (Remotely Operated Vehicle), has emerged as a wise and efficient alternative [2, 3]. These robotic systems typically rely on various sensors, such as imaging sonar, magnetic sensors, and cameras, to enhance operational efficiency and ensure safety. Among these, underwater cameras are particularly valuable due to their ability to achieve high precision in short-range target detection and to capture rich visual data from underwater objects [4]. Nevertheless, the underwater environment presents considerable challenges, including limited visibility, the absence of reliable reference points, and difficulties in signal transmission [5]. Under such harsh conditions, accurately identifying the category of a detected object can be difficult, particularly at longer detection ranges. Similarly, due to poor visibility, it is often challenging for the pilot to accurately judge the position of the detected object. It is necessary to develop a real-time display system capable of providing both the category and the position of the detected object when the detection range is reasonable. To enable the development of such a system, the simultaneous identification and localization of underwater objects must be achieved, where identification provides the category information and the localization offers the position data.

A. Underwater identification

Underwater identification aims to classify and recognise the category of underwater objects, which is typically carried out before the localisation. Relevant approaches can be classified into image processing, classification and deep learning techniques.

1) Image processing: Image processing techniques aim to carry out some necessary tasks on the image to prepare it for classification, for example, image enhancement, feature extraction, such as image segmentation and region of interest (ROI). Image processing is an important technique for computer vision, it has been widely applied in marine detection, food evaluation, and medical analysis [6, 7]. Some of the most relevant progress is in [8, 9, 10, 11, 12].

2) Classification: Classification is mainly realised by traditional machine learning methods. The transfer learning method is also included because it provides sufficient training data for machine learning. Some of the most relevant progress is in [13, 14, 15, 16, 17, 18].

3) Deep learning: Deep learning is a popular and up-to-date technique that combines the function of image processing (i.e. feature extraction) and classification. Thanks to the significant development in neural networks and their strong classification ability, it is considered the most popular method in the image classification and identification field. It has been applied in biomedicine, health management, and material [19, 20, 21]. Some of the most relevant progress is in [22, 23, 24, 25, 26, 27, 28, 29, 30].

B. Underwater localisation

Localisation of underwater structures is a common procedure in fields such as underwater surveillance, operations, maintenance, and measurement [31]. The localisation is carried out when the object is already identified. Relevant techniques include visual localisation, and visual marker.

1) Visual localisation: Visual localisation is a typical and common method in underwater localisation. It is a popular technique for short-range, precise localisation. They are widely applied in unmanned aerial and land vehicles, marine detection, and humanoid robot [32]. Some of the most relevant progress is in [33, 34, 35, 36, 37, 38].

2) Visual Marker: Visual marker is a popular passive localisation method that provides high-precision results within a short detection range. It is widely applied in localisation for various scenarios. Some of the most relevant progress is in [39, 40, 41, 42, 43].

The simultaneous and real-time identification and localisation of underwater objects is crucial for efficient underwater robotic operations. However, current research lacks integrated approaches that achieve both tasks simultaneously. Some studies focus solely on underwater identification using deep learning and machine learning methods but overlook localisation techniques. Conversely, other works emphasise underwater localisation, employing sensors such as imaging sonar, electromagnetic sensors or underwater acoustic positioning systems, while neglecting identification. Furthermore, published papers only provide limited resources on real-time identification or real-time localisation. This paper introduces a method that provides operation-related information for ROV pilots. The proposed approach processes video from the onboard camera to identify object categories and estimate their distances. This is accomplished through simultaneous, real-time identification and localisation. In this way, the system provides operational instructions to the pilot, thereby facilitating smoother robotic control. Identification is performed using Detectron2 [44], leveraging the Faster RCNN algorithm, while localisation is achieved through either a deep learning-based depth estimation model or a camera imaging principle-based method. The identification and localisation are achieved simultaneously by using the identification method to guide the localisation. To ensure real-time performance, the system is deployed on Google Colab, which provides the necessary computational resources for fast processing. Ultimately, the proposed system contributes to research in marine robotics, laying a solid foundation for marine robotic automation.

II. METHODOLOGY

The methodology of this study includes underwater object identification and localisation. Mooring chains, which are common subsea objects, are used in this study to test the identification and localisation approach proposed. The methodology aims to realise simultaneous, real-time identification and localisation of the considered structure.

A. Underwater identification

Underwater identification provides the category of the detected object. In this study, it is defined as category definition and object labelling.

Category definition:

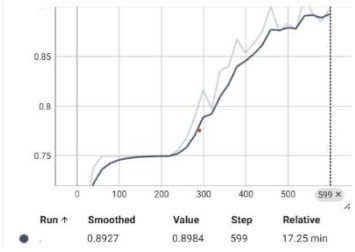
Category definition is the basis of the classification of the detected object. In this study, the employed algorithm aims to determine the category of the underwater structure exclusively using an underwater camera. Deep learning-based category definition is adopted in this study, as it offers better performance compared to traditional machine learning. This is because deep learning methods can learn from their errors and automatically resolve them, while machine learning often requires human supervision [45].

The deep learning algorithms used for category definition can be categorised into one-stage and two-stage algorithms [46]. Two-stage algorithms, such as those from the RCNN series, generally offer higher precision but require more computational power. On the other hand, one-stage algorithms, such as the YOLO series, are faster but may sacrifice some accuracy.

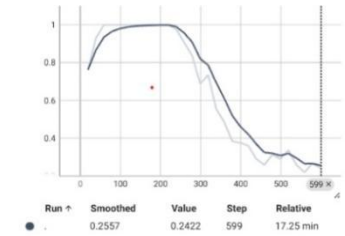
To achieve higher identification precision, this study applies and describes the two-stage algorithm Faster RCNN. Specifically, the 'Faster RCNN X101-FPN' model from Detectron2 is selected due to its high precision while maintaining decent inference speed. Relevant information about this model is provided in [47].

Before identification, model training must be performed using a dataset different from the actual test data. Various training datasets are available online. The dataset used in this study is obtained from Roboflow, a platform known for hosting the world's largest collection of open-source computer vision datasets and APIs [48]. It offers training datasets for a wide variety of objects, making it convenient to find the desired dataset. For this study, the dataset named 'Chain link' on Roboflow is used for training [49]. It includes high-quality images of chain links in different scenarios, which is suitable for the identification of mooring chains.

To ensure model reliability, validation is conducted during training. Validation is a technique in deep learning that uses a separate validation dataset to evaluate model performance. Moreover, training iterations significantly influence the quality of the identification model; therefore, it is important to determine an approximately optimal number of training iterations. The accuracy and false negative results for the Faster RCNN X101-FPN model are shown in Figure 1.



(a) cls accuracy (classification accuracy)



(b) false negative

Figure 1. Results of the accuracy and false negative plots for Faster RCNN X101-FPN under 600 training iterations.

From the results in Figure 1, it can be observed that the model's accuracy approaches 0.9, with the accuracy plot gradually levelling off as the training iterations reach 600, indicating satisfactory performance. The false negative rate initially increases up to 300 iterations but then drops sharply, and similarly begins to converge around 600 iterations. These trends suggest that 600 is an appropriate choice for the number of training iterations, offering both good and stable performance.

To further evaluate the model, a testing procedure is conducted using input images that were not included in the training phase. This process assesses the model's performance by analysing its identification results on unseen data. The testing output includes the bounding box, confidence score, and object category: the bounding box indicates the region where the object is detected, the confidence reflects how certain the model is about the detected object's category, and the category specifies the identified class.

Object labelling:

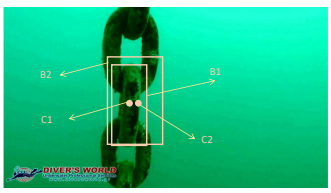
Object labelling is a procedure within the identification process that enables the operator to achieve localisation. In the context of mooring chains, labelling individual chain links is crucial for identifying a specific link, as each detected chain link is assigned a unique number. This facilitates the operations around the mooring chain.

The procedures of object labelling are quite similar to those of object identification. However, the key difference between object labelling and category definition lies in the incorporation of identification false-prevention methods in the former. These methods—including distance filtering, abnormal object filtering, and detected object recalling—are implemented to retain correct detections and eliminate false ones, thereby leading to smoother and more reliable labelling results.

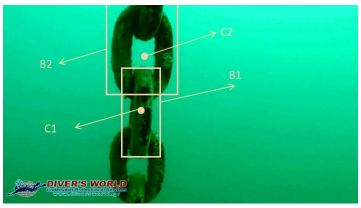
At the beginning of detection, the first detected bounding box is always considered a valid object. The centre position of this bounding box is calculated and denoted as (x_1, y_1). When a second bounding box appears, its centre is calculated as (x_2, y_2). A distance threshold d_1 is defined between the two centres, and this spatial relationship can be expressed using Equation (1):

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \leq d_1 \qquad (1)$$

If Equation (1) is satisfied—that is, the distance between the centres of two successive bounding boxes is within a predefined threshold—the newly detected object is considered identical to the previously detected one. Otherwise, it is classified as a new object. This concept is illustrated in Figure 2, where B1 represents Bounding box 1, C1 represents Center 1, B2, C2 are similar to that of B1 and C1.



(a) Case 1: Two bounding boxes are regarded as one object.



(b) Case 2: Two bounding boxes are regarded as different objects.

Figure 2. Diagram of the distance filtering.

During detection, the model may generate false positives by mistakenly identifying non-object regions as valid targets, which increases the number of labelled objects inaccurately. To mitigate this, a temporal consistency check is implemented: a detected object must be observed in a set number of consecutive frames before it is confirmed and assigned a unique label. This process improves the reliability and stability of object labelling. Moreover, objects that have already been identified may temporarily disappear from detection due to occlusions, environmental factors, or sensor noise. To handle such cases and maintain consistent labelling, a technique known as detected object recalling is employed. This method matches newly detected bounding boxes with previously identified objects based on spatial proximity. If the centre of a new detection lies within a specified distance of a known object's center, the original identification number is reassigned, ensuring continuous and accurate tracking of objects over time.

B. Underwater localisation

Underwater localisation aims to determine the position of the detected object relative to the camera. In this study, two approaches are employed and compared: Method 1, which utilises deep learning, and Method 2, which relies on vision-based geometric analysis of the detected object.

Method 1:

Method 1 utilises the Metric 3D algorithm, a high-quality and robust deep learning approach for distance estimation using a monocular camera [50]. This algorithm processes the input image to predict its depth map, from which the distance (in meters) of each pixel can be determined. In this context, the distance refers specifically to the separation between the underwater camera and the detected object. By leveraging these distance measurements, the relative position of the detected object with respect to the camera is obtained, thereby achieving localisation.

The Metric 3D framework provides pre-trained prediction models that can be readily utilized by referencing its implementation. The commonly available pre-trained models include 'metric3d_vit_small', 'metric3d_vit_large', and 'metric3d_vit_giant2', all originally developed for image classification tasks [50]. Generally, larger models offer higher accuracy in distance prediction, albeit with increased computational demands. In practice, the 'metric3d_vit_small' model is often sufficiently precise for underwater localisation applications.

For localisation purposes, an identification model must first be established to generate bounding boxes around detected objects. These bounding boxes are subsequently used to extract distance information from the predicted depth map. Once both the depth map and bounding boxes are obtained, distance data for each pixel within the

bounding box can be analysed. Specifically, the distance from the camera to a chain link is determined by calculating the minimum distance value within the bounding box. This approach effectively addresses the presence of hollow regions in the chain link structure, which can produce abnormally large distance measurements, by minimising their impact on localisation accuracy.

The primary advantage of this method lies in its independence from prior knowledge of the object's physical dimensions—a significant benefit given the challenges associated with measuring underwater structures. However, the method also entails considerable computational complexity, as it relies on two deep learning algorithms and thus requires a powerful GPU to ensure real-time performance.

Method 2:

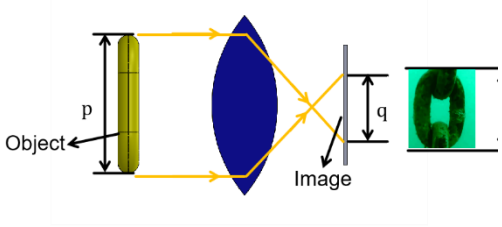
Although Method 1 offers a broadly applicable localisation solution, it imposes significant demands on hardware resources such as GPUs. To address the need for a computationally efficient alternative, Method 2 is proposed. This method is grounded in trigonometric principles and leverages known object dimensions to calculate both the distance and the visual angle between the camera and the detected object. The approach applies fundamental concepts of camera imaging and geometric similarity. The corresponding formulas used to compute the distance d and visual angle α are presented in Equations (2), (3), and (4) below:

$$\frac{\mathrm{d}}{f} = \frac{p}{q} \tag{2}$$

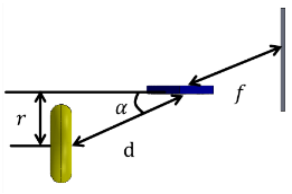
$$\mathrm{d} = \frac{\mathrm{fp}}{q} \tag{3}$$

$$\alpha = \sin^{-1}\frac{\mathrm{r}}{\mathrm{d}} \tag{4}$$

Where d is the distance between the camera and the object, f is the focal length of the camera, p is the height of the chain, which is obtained before the detection, q is the height of the chain link in the sensor, and r is the chain deviation from the central line. The height of the chain in the sensor q and the chain deviation from the central line r are obtained during the detection. Figure 3 illustrates the principle of Method 2.



(a) Front view



(b) Top view

Figure 3. The principle of Method 2.

It is assumed that the centre of the chain link and the centre of the camera lens lie on the same horizontal plane, and consequently, the calculated angle is confined to this horizontal plane without accounting for any vertical deviation. Before image processing, essential parameters from both the camera and the object must be obtained. Specifically, the camera parameters required include the focal length and the pixel size, while the physical dimensions of the object must also be measured or otherwise acquired. In this study, the object size is determined either by measuring the chain dimensions from a digital file or by referencing specifications provided in the product description. It is important to note that the apparent width of the chain link may vary during detection, as the camera can capture the chain link from two distinct viewpoints, which are illustrated in Figure 4.

(a) Front view (b) Left view

Figure 4. The possible views of the chain link

Since the height of the chain link remains constant during the movement of the underwater robot, this dimension is utilised for distance calculation, corresponding to the parameter p in Equations (2) and (3). Similar to Method 1, an identification model must first be established to generate bounding boxes around detected objects. The dimensions of these bounding boxes are subsequently used to estimate the distance to the detected object. Specifically, through category definition, the height of the bounding box is regarded as the height of the chain link q in Equations (2) and (3). Additionally, the lateral deviation of the chain from the central axis, denoted as r in Equation (4), is computed by measuring the distance between the bounding box centre and the image's central vertical line. Using these parameters, the distance d is calculated in accordance with Equation (3).

The principal advantage of Method 2 lies in its substantially reduced computational burden, as it employs only a single deep learning model. Furthermore, this approach provides richer information compared to Method 1. A summary of the data obtained through both Method 1 and Method 2 is presented in Table 1.

Table 1. The information obtained within Method 1 and Method 2.

Name	Information obtained	Description
Method 1	Distance (Depth)	The distance (depth) represents the distance between the object center and the camera
Method 2	Distance	Identical to Method 1
	Angle	The calculated angle represents the horizontal angle

From Table 1, it can be inferred that Method 2 may acquire more information than Method 1, especially when the dimension of the detected object is not known in Method

1. However, this method needs to know the dimension of the object and the relevant parameters of the camera in advance, which is difficult to implement when not all dimensions of the object can be obtained underwater. In addition, the detection position is rather restricted because there is a special positional relationship between the camera lens and the chain link.

C. Implementation:

Underwater navigation is supported through the deployment of onboard cameras, which continuously capture and transmit video streams for subsequent computational analysis. To perform object identification, labelling, and localisation, the recorded video is initially decomposed into individual image frames. Each frame is then sequentially processed using a pre-trained identification model capable of detecting and classifying objects. When localisation is required, the corresponding localisation algorithm is also applied to each frame to extract spatial information.

Upon completion of the frame-wise analysis, the processed images are recompiled to reconstruct a comprehensive output video that visually communicates the results. The specific content of the resulting video varies according to the task objective. For object identification, the video displays the classification of each detected object along with its associated confidence score. In object labelling tasks, the video additionally includes unique identification numbers to maintain consistent tracking of objects across frames. For underwater localisation, the visual output is determined by the chosen localisation method: Method 1 displays object classification and estimated distance, whereas Method 2 additionally provides angular deviation with respect to the camera's optical axis.

This integrated visualisation approach ensures the delivery of complete semantic and spatial context, thereby enhancing the reliability and interpretability of underwater inspection and navigation systems.

III. Experiment

To verify the feasibility of the proposed identification and localisation methods, experiments were conducted in an underwater environment to simulate realistic operating conditions. During the experiments, a ROV equipped with an underwater camera was used to navigate and capture continuous video footage of the chain link, providing the necessary data for processing. The focus of the experiments was on acquiring object category information and estimating object locations based on the video data collected by the ROV. The recorded footage was subsequently uploaded to Google Colab, where it was processed using the developed identification and localisation procedure.

A. Experiment setup:

The experiment aims to achieve simultaneous identification and localisation of the chain link in an underwater environment. To support this objective, the materials and equipment used in the experimental setup are listed in Table 2.

Table 2. The material list for the experiment.

Name	Number	Description
Chain link	1	3D-printed using PETG to simulate a real chain link
Weight	1	The weight will relate to chain link and is placed at the bottom of water to prevent floating of chain link
ROV	1	Serves as the underwater mobile platform for video capture and manipulation
Laptop	1	Used for real-time data processing and to guide the pilot's operation

The experiment setup is organised as shown below:

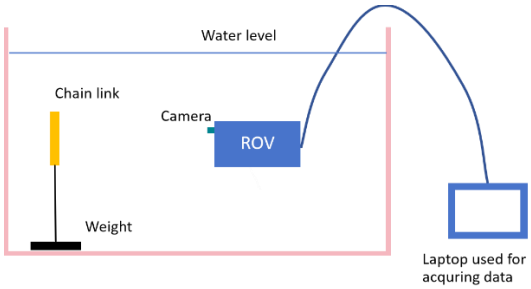


Figure 5. The experiment setup.

B. Experiment procedure:

The experiment is conducted based on the setup illustrated in Figure 5. First, a ruler is affixed to the side of the experiment tank to enable measurement of the approximate distance between the ROV and the chain link. Next, the pilot manoeuvres the ROV toward the chain link, while another participant records the time, approximate distance, and the distance estimated by the identification and localisation system throughout the ROV's movement. Finally, the experiment is repeated multiple times to collect comprehensive data, from which the localisation error is calculated and analysed.

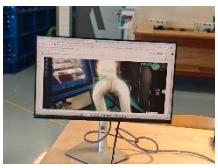
IV. Results and Discussions

Throughout the experiment, the pilot first controls the ROV to approach the chain link gradually. Then, the ROV returns to its original place. This process is repeated several times to collect enough experimental data. Figure 6 shows some important scenes during the experiment.

(a) Approaching the chain link

(b) Clamping the chain link

(c) Video data acquisition



(d) Online platform processing

Figure 6. Some important scenes during the experiment.

As the ROV approaches the chain link, a visual example of successful identification and localization is presented in Figure 7.

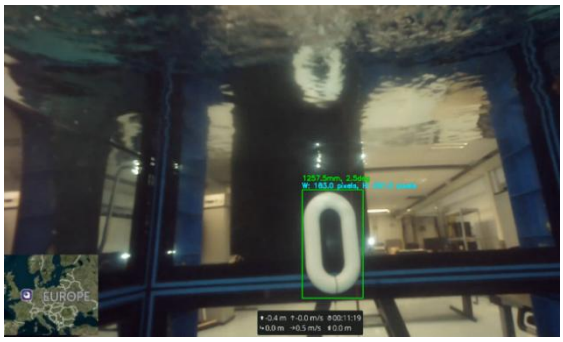


Figure 7. Visual example of successful identification and localisation.

The experimental video demonstrates that the system is capable of accurately identifying and localizing the chain link, particularly when it is closer to the ROV's camera. However, occasional detection failures still occur, suggesting that the identification model could benefit from further refinement. To evaluate the localization performance, the approximate measured distances and those estimated by the localization system are compared, as shown in Table 3.

Table 3. The measured distance and the distance obtained by the localization program in the experiment.

Order	Estimated distance (m)	Obtained distance from the program (m)	Error
1	1.0	1.2	20.0%
2	1.3	1.5	15.4%
3	1.0	1.3	30.0%
4	0.6	0.62	3.3%
5	1.5	1.2	20.0%
6	0.7	0.65	7.1%
7	2.0	2.0	0
8	1.45	1.5	3.4%
9	2.0	1.6	20.0%
10	0.9	1	11.1%

The recorded data showed that the estimated distances deviated from the actual measurements by no more than 30%. This level of accuracy may be sufficient to support effective remote operations. Deviations were primarily caused by occasional Internet transmission delays during video upload and processing. Additionally, imperfect bounding box detection, particularly when the chain link was partially occluded or appeared further from the camera may have introduced slight inconsistencies in real-time analysis. These results suggest that the proposed system can provide actionable distance feedback to guide ROV operations, especially in structured underwater environments where target objects are relatively static and predictable in size and appearance. This supports the method's feasibility for real-world deployment in controlled scenarios.

It is also worth noting that only Localization Method 2 was employed during the experiment. Despite being a simpler and more efficient approach compared to Method 1, it achieved comparable localisation accuracy while requiring significantly fewer computational resources, indicating that Method 2 may be sufficient to meet the experimental requirements.

V. CONCLUSIONS

This study explored an integrated system for underwater object identification and localisation using ROV-acquired video data, aimed at enabling safer and more efficient underwater vehicle operations. The system combined a deep learning-based object detection model with two localisation approaches to provide real-time distance estimation during underwater operations.

The object identification module employed a convolutional neural network trained on labelled underwater images. It detected chain links in most scenarios, generating bounding boxes that served as inputs for localisation. While occasional detection failures occurred due to internet delay, turbidity, reflections, or partial occlusion, the model demonstrated strong performance when the chain was fully visible.

Two localisation methods were evaluated. Method 1 used pixel-wise depth estimation within the bounding box to calculate the average object distance. It required no prior knowledge of object dimensions or camera parameters, offering high adaptability at the cost of greater computational demand. Method 2, in contrast, estimated distance using the bounding box height, known object dimensions, and camera intrinsics. This approach proved more efficient and better suited for real-time applications, whose detection accuracy remained high.

Experiments demonstrated the system's effectiveness in underwater conditions. As the ROV approached the chain link, the system consistently performed simultaneous detection and localisation. Distance estimations deviated by no more than 30% from actual measurements, with a greater likelihood of successful identification and localisation when the object was positioned closer to the camera.

In summary, the proposed method is considered meeting sufficient requirements for real-time underwater object localisation, especially in structured scenarios with known object and camera parameters. It holds potential for future applications in underwater exploration and operation.

REFERENCE

[1] Bennett, P. B. (1989). Physiological limitations to underwater exploration and work. Comparative Biochemistry and Physiology Part A: Physiology, 93(1), 295-300.

[2] Teague, J., Allen, M. J., & Scott, T. B. (2018). The potential of low-cost ROV for use in deep-sea mineral, ore prospecting and monitoring. Ocean Engineering, 147, 333-339.

[3] Macreadie, P. I., McLean, D. L., Thomson, P. G., Partridge, J. C., Jones, D. O., Gates, A. R., ... & Fowler, A. M. (2018). Eyes in the sea: unlocking the mysteries of the ocean using industrial, remotely operated vehicles (ROVs). *Science of the Total Environment*, *634*, 1077-1091.

[4] Cardaillac, A., & Ludvigsen, M. (2023). Camera-sonar combination for improved underwater localization and mapping. IEEE Access, 11, 123070-123079.

[5] Merveille, F. F. R., Jia, B., & Xu, Z. (2024). Advancements in Underwater Navigation: Integrating Deep Learning and Sensor Technologies for Unmanned Underwater Vehicles. Preprints.

[6] Du, C. J., & Sun, D. W. (2004). Recent developments in the applications of image processing techniques for food quality evaluation. Trends in food science & technology, 15(5), 230-249.

[7] Dougherty, G. (2009). Digital image processing for medical applications. Cambridge University Press.

[8] Liang, Z., Wang, Y., Ding, X., Mi, Z., & Fu, X. (2021). Single underwater image enhancement by attenuation map guided color correction and detail preserved dehazing. Neurocomputing, 425, 160-172.

[9] Cheng, H. D., Jiang, X. H., Sun, Y., & Wang, J. (2001). Color image segmentation: advances and prospects. Pattern recognition, 34(12), 2259-2281.

[10] Bosse, S., & Kasundra, P. (2022). Robust Underwater Image Classification Using Image Segmentation, CNN, and Dynamic ROI Approximation. Engineering Proceedings, 27(1), 82.

[11] Chen, Z., Zhang, Z., Dai, F., Bu, Y., & Wang, H. (2017). Monocular vision-based underwater object detection. Sensors, 17(8), 1784.

[12] Jian, M., Liu, X., Luo, H., Lu, X., Yu, H., & Dong, J. (2021). Underwater image processing and analysis: A review. Signal Processing: Image Communication, 91, 116088.

[13] Jian, M., Yang, N., Tao, C., Zhi, H., & Luo, H. (2024). Underwater object detection and datasets: a survey. Intelligent Marine Technology and Systems, 2(1), 9.

[14] Langner, F., Knauer, C., Jans, W., & Ebert, A. (2009, May). Side scan sonar image resolution and automatic object detection, classification and identification. In OCEANS 2009-EUROPE (pp. 1-8). IEEE.

[15] Dos Santos, M., Ribeiro, P. O., Núñez, P., Drews-Jr, P., & Botelho, S. (2017). Object classification in semi structured enviroment using forward-looking sonar. Sensors, 17(10), 2235.

[16] Luo, X., Chen, L., Zhou, H., & Cao, H. (2023). A survey of underwater acoustic target recognition methods based on machine learning. Journal of Marine Science and Engineering, 11(2), 384.

[17] Chen, L., Huang, Y., Dong, J., Xu, Q., Kwong, S., Lu, H., ... & Li, C. (2024). Underwater Object Detection in the Era of Artificial Intelligence: Current, Challenge, and Future. arXiv preprint arXiv:2410.05577.

[18] Ge, H., Dai, Y., Zhu, Z., & Liu, R. (2022). A deep learning model applied to optical image target detection and recognition for the identification of underwater biostructures. Machines, 10 (9), 809.

[19] Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. Mechanical Systems and Signal Processing, 107, 241-265.

[20] Mamoshina, P., Vieira, A., Putin, E., & Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. Molecular pharmaceutics, 13(5), 1445-1454.

[21] Choudhary, K., DeCost, B., Chen, C., Jain, A., Tavazza, F., Cohn, R., ... & Wolverton, C. (2022). Recent advances and applications of deep learning methods in materials science. npj Computational Materials, 8(1), 59.

[22] Li, Y., Lu, H., Li, J., Li, X., Li, Y., & Serikawa, S. (2016). Underwater image de-scattering and classification by deep neural network. Computers & Electrical Engineering, 54, 68-77.

[23] Xu, S., Zhang, M., Song, W., Mei, H., He, Q., & Liotta, A. (2023). A systematic review and analysis of deep learning-based underwater object detection. Neurocomputing, 527, 204-232.

[24] Chen, L., Huang, Y., Dong, J., Xu, Q., Kwong, S., Lu, H., ... & Li, C. (2024). Underwater Object Detection in the Era of Artificial Intelligence: Current, Challenge, and Future. arXiv preprint arXiv:2410.05577.

[25] Er, M. J., Chen, J., Zhang, Y., & Gao, W. (2023). Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review. Sensors, 23(4), 1990.

[26] Guntha, P., & Beaulah, P. M. R. (2024, April). A Comprehensive Review on Underwater Object Detection Techniques. In 2024 International Conference on Computing and Data Science (ICCDS) (pp. 1-6). IEEE.

[27] Cheng, N., Xie, H., Zhu, X., & Wang, H. (2023). Joint image enhancement learning for marine object detection in natural scene. Engineering Applications of Artificial Intelligence, 120, 105905.

[28] Zhu, P., Isaacs, J., Fu, B., & Ferrari, S. (2017, December). Deep learning feature extraction for target recognition and classification in underwater sonar images. In 2017 IEEE 56th annual conference on decision and control (CDC) (pp. 2724-2731). IEEE.

[29] Gašparović, B., Lerga, J., Mauša, G., & Ivašić-Kos, M. (2022). Deep learning approach for objects detection in underwater pipeline images. Applied artificial intelligence, 36(1), 2146853.

[30] Yu, Y., Zhao, J., Gong, Q., Huang, C., Zheng, G., & Ma, J. (2021). Real-time underwater maritime object detection in side-scan sonar images based on transformer-YOLOv5. Remote Sensing, 13(18), 3555.

[31] Fareh, R., Khadraoui, S., Abdallah, M. Y., Baziyad, M., & Bettayeb, M. (2021). Active disturbance rejection control for robotic systems: A review. Mechatronics, 80, 102671.

[32] Boittiaux, C., Dune, C., Ferrera, M., Arnaubec, A., Marxer, R., Matabos, M., ... & Hugel, V. (2023). Eiffel Tower: A deep-sea underwater dataset for long-term visual localization. The International Journal of Robotics Research, 42(9), 689-699.

[33] Trslić, P., Weir, A., Riordan, J., Omerdic, E., Toal, D., & Dooly, G. (2020). Vision-based localization system suited to resident underwater vehicles. Sensors, 20(2), 529.

[34] Nuske, S., Roberts, J., Prasser, D., & Wyeth, G. (2010, July). Experiments in visual localization around underwater structures. In Field and Service Robotics: Results of the 7th International Conference (pp. 295-304). Berlin, Heidelberg: Springer Berlin Heidelberg.

[35] Xanthidis, M., Joshi, B., Roznere, M., Wang, W., Burgdorfer, N., Li, A. Q., ... & Rekleitis, I. (2022, September). Towards mapping of underwater structures by a team of autonomous underwater vehicles. In The International Symposium of Robotics Research (pp. 170-185). Cham: Springer Nature Switzerland.

[36] dos Santos, M. M., de Oliveira Evald, P. J. D., De Giacomo, G. G., Drews-Jr, P. L. J., & da Costa Botelho, S. S. (2023). A probabilistic underwater localization based on cross-view and cross-domain acoustic and aerial images. Journal of Intelligent & Robotic Systems, 108(3), 34.

[37] Zhong, L., Li, D., Lin, M., Lin, R., & Yang, C. (2019). A fast binocular localization method for AUV docking. Sensors, 19(7), 1735.

[38] Qin, J., Li, M., Li, D., Zhong, J., & Yang, K. (2022). A survey on visual navigation and positioning for autonomous UUVs. Remote Sensing, 14(15), 3794.

[39] Buchan, A. D., Solowjow, E., Duecker, D. A., & Kreuzer, E. (2017, September). Low-cost monocular localization with active markers for micro autonomous underwater vehicles. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4181-4188). IEEE.

[40] Chavez, A. G., Mueller, C. A., Doernbach, T., & Birk, A. (2019). Underwater navigation using visual markers in the context of intervention missions. International journal of advanced robotic systems, 16(2), 1729881419838967.

[41] Negre, A., Pradalier, C., & Dunbabin, M. (2008). Robust vision-based underwater homing using self-similar landmarks. Journal of Field Robotics, 25(6-7), 360-377.

[42] Wei, Q., Yang, Y., Zhou, X., Fan, C., Zheng, Q., & Hu, Z. (2023). Localization method for underwater robot swarms based on enhanced visual markers. Electronics, 12(23), 4882.

[43] Folkesson, J., Leederkerken, J., Williams, R., Patrikalakis, A., & Leonard, J. (2008). A feature based navigation system for an autonomous underwater robot. In Field and Service Robotics: Results of the 6th International Conference (pp. 105-114). Springer Berlin Heidelberg.

[44] Detectron 2. Detectron2 Model Zoo and Baselines. From: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md

[45] Google Cloud. Deep learning & machine learning. From: https://www.google.com/search?q=deep+learning+methods+can+learn+from+their+own+errors%2C+while+machine+learning+often+requires+human+intervention&rlz=1C1ONGR_nlNL1073NL1073&oq=deep+learning+methods+can+learn+from+their+own+errors%2C+while+machine+learning+often+requires+human+intervention&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDc5M2owajE1qAIIsAIB8QWVp1RDYfWolw&sourceid=chrome&ie=UTF-8

[46] Kang, J., Tariq, S., Oh, H., & Woo, S. S. (2022). A survey of deep learning-based object detection methods and datasets for overhead imagery. IEEE Access, 10, 20118-20134.

[47] Detectron2 Model Zoo and Baselines. From: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md

[48] Roboflow. From: https://universe.roboflow.com

[49] Roboflow. Chain link. From: https://universe.roboflow.com/university-of-pennsylvania/chain-link

[50] Metric3D Project. From: https://github.com/YvanYin/Metric3D

B. Grabber mechanism

The grabber mechanism is designed to enable secure fixation between the underwater robotic system and the underwater chain link. Mounted on the robot, the grabber must be flexible enough to deform under the robot's thrust force while maintaining sufficient rigidity to withstand currents and wave forces. The operational procedure of the grabbing mechanism is illustrated in Figure 71.

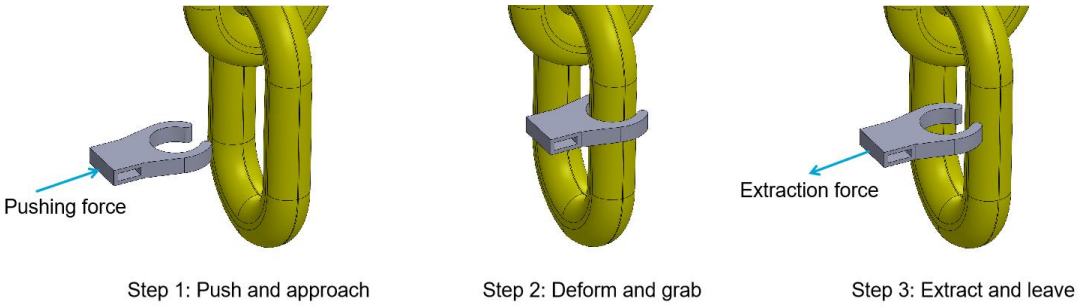


Figure 71. Functioning procedures of grabbing mechanism.

Initially, the grabber aligns with and approaches the chain link. The robot then applies thrust force to deform the grabber, allowing it to grasp the chain. Once engaged, the grabber firmly holds the chain link, stabilizing the entire system relative to the chain. When release is required, the robot reverses the thrust force to disengage and withdraw the grabber.

Since the thrust force generated by the underwater robot is relatively limited, the grabber material must balance flexibility—enabling deformation under thrust—and rigidity—resisting environmental forces. This balance can be achieved either by selecting an inherently flexible material or by adjusting the grabber's flexibility dynamically during operation. Thermoplastic polyurethane (TPU) is a flexible material with easily adjustable flexibility, while polylactic acid (PLA) offers flexibility control through environmental temperature changes, which can be adjusted during operation. Based on these properties, TPU and PLA were chosen as candidate materials for the grabber. To minimize costs, the grabbers were fabricated using 3D printing technology. Figure 72 shows the 3D-printed TPU and PLA components used in this study.

(a) The 3D printed TPU piece [146].

(b) The 3D printed PLA pieces [147].

Figure 72. The 3D printed TPU and PLA pieces.

To further introduce the characteristics of TPU and PLA, some relevant parameters and the description is indicated in Table 10.

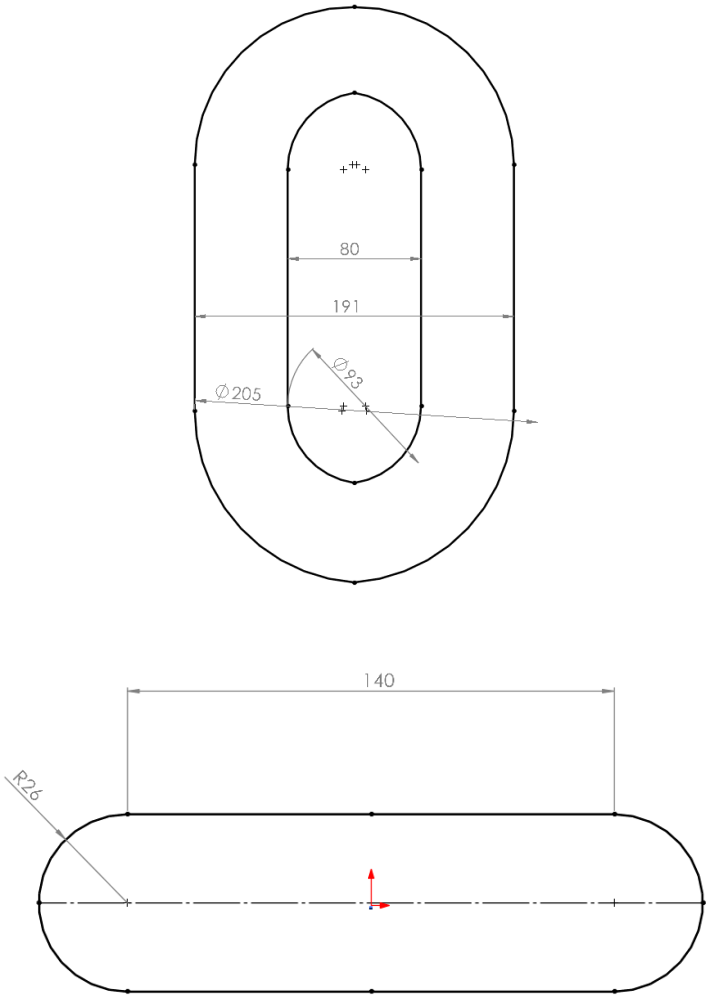
Table 10. Some relevant parameters and the description of TPU and PLA.

Name	Description	Young's modulus	Yield strength
TPU	High elasticity, wear resistance, and resilience. A popular flexible 3D printing material effective in extremely low temperatures	0.16 GPa	29.3 MPa
PLA	One of the most eco-friendly options for 3D printer and one of the most dominant material the 3D printing industry	4.1 Gpa	26.1 MPa

TPU grabber

Common materials are typically too rigid to deform effectively under the forces exerted by underwater robotics such as ROVs. TPU, however, is a flexible material that can easily deform in response to external underwater forces. Additionally, its flexibility can be adjusted by modifying the design dimensions, making TPU a suitable candidate material for the grabber. In the experiment, the TPU grabber must deform under the ROV's thrust force to clamp the chain securely and maintain stability. This requires the grabber to be sufficiently thin to allow deformation, yet once flexible enough, it should also be thick enough to resist external forces from currents and waves. To ensure effective clamping, the TPU grabber's dimensions were designed to match those of the target chain. Since a real chain link may be too large and heavy for the experimental setup, a scaled-down 3D-printed replica of the chain link was used instead. The image of this 3D-printed chain link is shown in Figure 73.

(a) The chain link



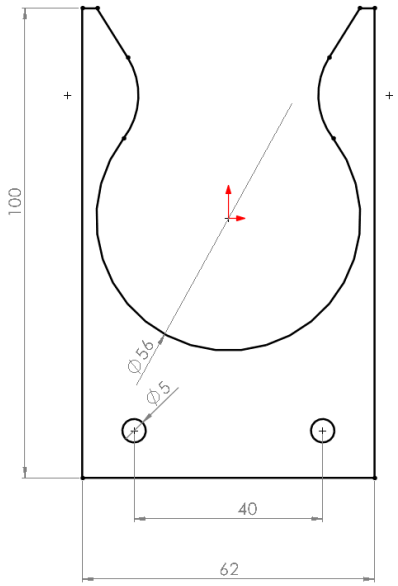
(b) Dimensions of the chain link

Figure 73. The chain link.

From Figure 73 (b), it can be inferred that the inner diameter of the TPU grabber should be approximately 55 mm to ensure a stable grip on the chain link. The TPU

grabber is shown in Figure 74.

(a) The TPU grabber



(b) Dimensions of the TPU grabber (The thickness is 30mm)

Figure 74. The TPU grabber

To verify the feasibility of the design, a simulation test was conducted where the TPU grabber attempts to clamp the chain link. It was observed that the grabber must deform under a certain force to successfully engage with the chain. Once fully clamped, an external force is required to retract the grabber from the chain, demonstrating effective fixation. These results confirm that the TPU grabber design functions as intended. The clamping interaction between the TPU grabber and the chain link is illustrated in Figure 75.

Figure 75. The clamping between the chain link and the TPU grabber.

PLA grabber

Introduction

The PLA grabber is rigid at room temperature and requires heating to increase its flexibility and reduce rigidity. It was experimentally confirmed that applying a heating pad to the surface of the PLA grabber generates sufficient heat to enable deformation, allowing the chain link to be inserted. However, in underwater environments, the effectiveness of the heating pad is significantly diminished if not properly insulated from water exposure. Therefore, it is necessary to encapsulate the heating pad within the grabber structure to ensure efficient heat transfer and maintain the desired softening effect. The design of the PLA grabber with integrated heating pad is illustrated in Figure 76.

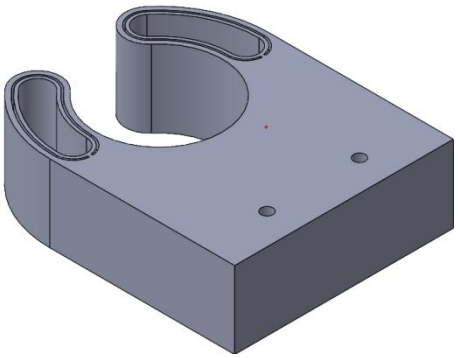


Figure 76. The design of the grabber.

The heating pad is positioned within a slot formed between the main body of the grabber and an internal hollow structure. It is designed to fit precisely within this slot, ensuring uniform heat distribution across the entire area, which facilitates consistent

softening of the grabber. Once placed, an adhesive is applied to both sides of the slot to seal the heating pad securely inside the grabber. Figure 77 shows the heating pad installed within the slot of the grabber.

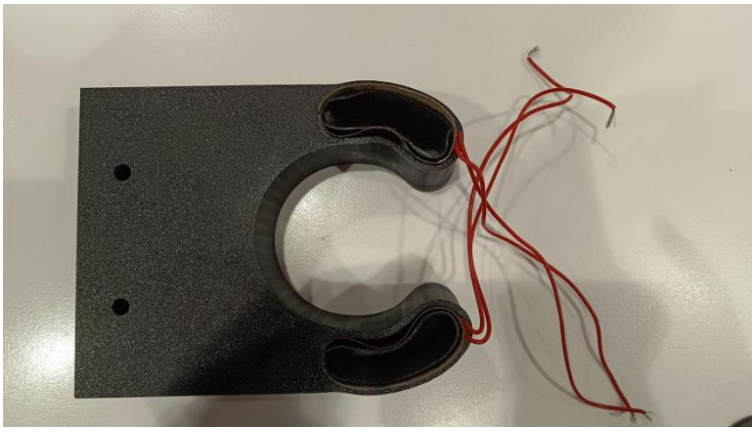


Figure 77. The heating pad in the slot of grabber.

Experimental setup

The PLA grabber's softening under heat is critical to enabling its deformation under external force, thereby allowing the chain link to be smoothly clamped. To validate this property, deformation experiments were conducted at room temperature. Prior to the experiments, the setup and procedures are described to provide a clear understanding of the experimental requirements. The materials used in the experiments are listed in Table 11.

Table 11. The required material list in PLA grabber experiment

Name	Number	Description
PLA grabber	1	The component softens when heated and hardens upon cooling. It functions as the gripping element for securing the chain.
Heating pad	2	These components are embedded within the grabber and heat up when voltage is applied.
Power source	1	A power source supplies voltage to the heating pad.
Cylindrical bar	1	A large object is used as a substitute for the chain link; this bar corresponds to the vertical segment of the chain link

The picture of the experimental setup is shown in Figure 78.

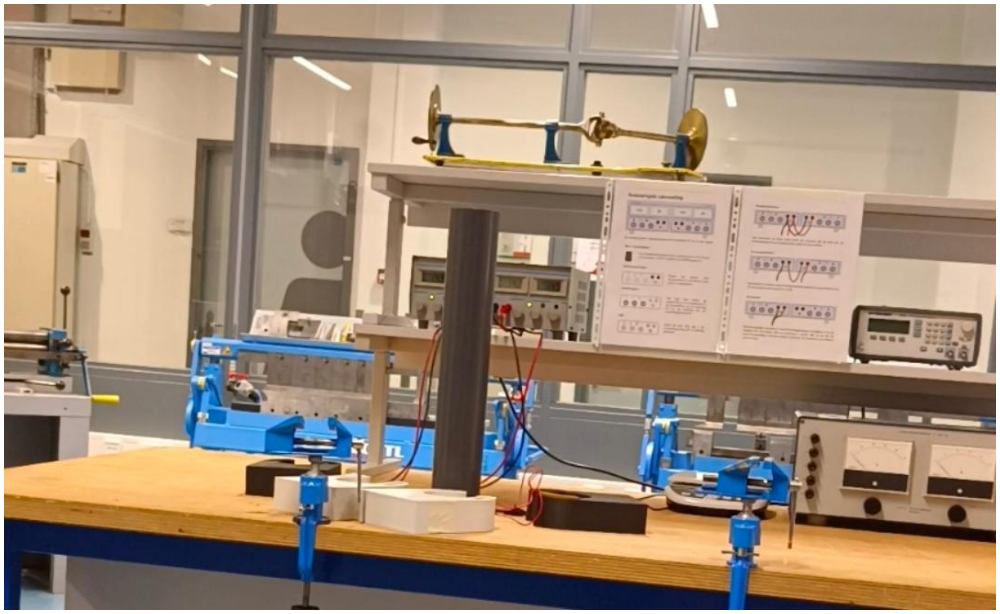


Figure 78. the experimental setup of PLA grabber. The black PLA grabber is equipped with heating pad, the bar will be clamped by the grabber, the power source behind the bar supplies electricity

Experiment

Once the experimental setup is complete, the formal experiment can commence. The key procedures are described as follows:

To initiate the experiment, the heating pad wires are connected to a power source to supply voltage. The nominal voltage for the heating pad is 18 V. At this voltage, the 3 mm-thick PLA grabber becomes highly flexible. Even at 15 V, the PLA grabber exhibits significant softening. The flexibility of the PLA grabber at 15 V is illustrated in Figure 79.

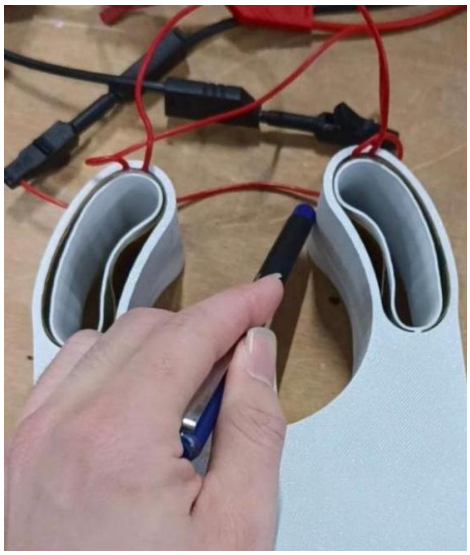


Figure 79. The softness of the PLA grabber when 15V voltage is applied on the heating pad

From Figure 79, it is evident that the right section of the grabber deforms readily

under a small compressive force applied with a pen, demonstrating considerable flexibility at 15 V. It can be inferred that increasing the voltage to the nominal 18 V would further enhance the grabber's softness, allowing for adjustment in flexibility depending on operational requirements.

Following the softening phase, the grabber is ready to accommodate the chain. In the experimental setup, a cylindrical bar, used as a surrogate for the chain link, can be inserted into the grabber with ease. The insertion of the cylindrical bar into the grabber is shown in Figure 80.

Figure 80. The entrance of the cylindrical bar to the grabber.

In Figure 80, it can be observed that the grabber has partially enclosed the cylindrical bar, surrounding approximately half of its circumference. During the experiment, the grabber successfully engaged the bar without requiring significant force, indicating that heating effectively facilitates the smooth insertion of the cylindrical bar into the grabber.

The successful insertion, however, is not the conclusion of the experiment. To securely hold the bar, the grabber must regain rigidity, which is achieved through cooling. Once the heating pad is turned off, the grabber gradually returns to room temperature and hardens. After cooling, the grabber is capable of firmly holding the heavy cylindrical bar, demonstrating the success of the experiment under ambient conditions. The stable grip of the PLA grabber is shown in Figure 81.

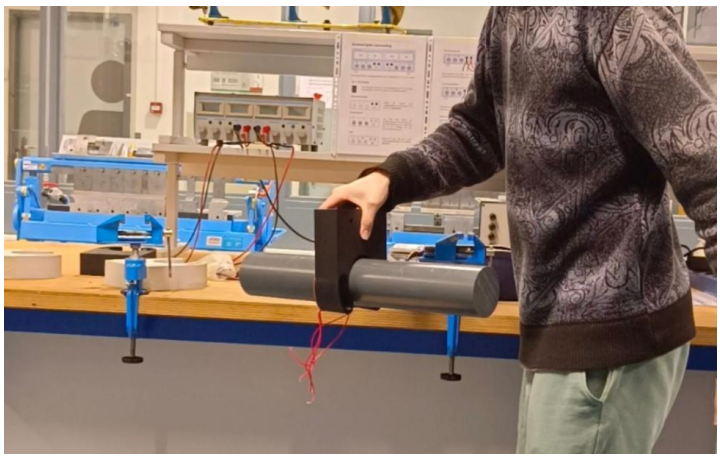


Figure 81. The stable grabbing of the PLA grabber.

This method offers the advantage of enabling the grabber to soften for easy entry of the chain link and subsequently harden to prevent potential detachment. Both deformation and re-hardening are conveniently controlled by the heating pad, which makes the approach well-suited for underwater scenarios. In such conditions, the chain can be more easily inserted into the grabber with the thrust from the ROV, while the hardened grabber can resist larger currents.

However, several challenges remain. First, irreversible thermal deformation may occur. While the deformation of the PLA grabber is generally elastic with sufficient material thickness, thinner sections may experience permanent deformation. For example, Figure 79 shows deformation in the grabber's inner structure under heating, which could reduce the effectiveness of heating and compromise structural stability. Second, the sealing effectiveness of the 3D-printed PLA parts is uncertain; it is unclear whether the grabber can fully isolate the heating pad from water intrusion. Lastly, the actual heating performance of the pad in underwater conditions requires further investigation to confirm its ability to adequately soften the grabber.

To address the second challenge, one potential improvement is designing a dedicated passage for the heating pad cables, minimizing their exposure to water and enhancing safety. This approach to improve the sealing of the PLA grabber is illustrated in Figure 82.

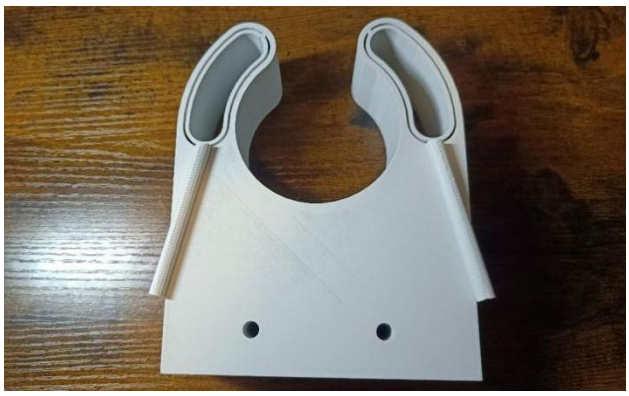


Figure 82. The method of improving the seal effect of the PLA grabber.

C. GUI

The Graphical User Interface (GUI) allows users to interact with electronic systems through intuitive graphical icons and visual components. In this project, the GUI is designed primarily to assist operators by providing clear situational awareness and data visualization. It facilitates and supports post-mission analysis and reflection. The following sections outline the procedures involved in the implementation of the GUI.

Underwater identification & Localization

To generate the data that feeds into the GUI, underwater object identification and localization are performed. The output is a processed video containing both positional and categorical information. Additionally, the positional data are saved in a text file, which serves as the data source for the GUI's visualization components. An overview of the saved data format is presented in Figure 83.

```
Frame 0: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 1: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 2: Avg Distance = 749.55 mm, Avg Angle = 0.02 degrees
Frame 3: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 4: Avg Distance = 808.51 mm, Avg Angle = 0.02 degrees
Frame 5: Avg Distance = 799.52 mm, Avg Angle = 0.02 degrees
Frame 6: Avg Distance = 736.51 mm, Avg Angle = 0.02 degrees
Frame 7: Avg Distance = 723.91 mm, Avg Angle = 0.02 degrees
Frame 8: Avg Distance = 781.29 mm, Avg Angle = 0.01 degrees
Frame 9: Avg Distance = 1228.09 mm, Avg Angle = 0.01 degrees
Frame 10: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 11: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 12: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 13: Avg Distance = 864.87 mm, Avg Angle = 0.01 degrees
Frame 14: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 15: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 16: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 17: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 18: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 19: Avg Distance = 1242.78 mm, Avg Angle = 0.01 degrees
Frame 20: Avg Distance = 1185.46 mm, Avg Angle = 0.01 degrees
Frame 21: Avg Distance = 1096.91 mm, Avg Angle = 0.01 degrees
Frame 22: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 23: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 24: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 25: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 26: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 27: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 28: Avg Distance = 1193.32 mm, Avg Angle = 0.00 degrees
Frame 29: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
Frame 30: Avg Distance = 0.00 mm, Avg Angle = 0.00 degrees
```

Figure 83. The overview of the data saved in a text document.

Load video

The GUI is integrated with the identification and localization modules, enabling the program to run once the underwater processing is complete. Upon launching the GUI, the initial interface appears blank, awaiting the user to load the processed video to enable data visualization. Figure 84 illustrates the layout of the initialized page.

Figure 84. The overview of the initialized page.

The interface is divided into two main zones: the functional zone, which contains all buttons and the progress bar, and the data zone, where relevant data will be displayed (detailed later). Initially, all buttons in the functional zone are disabled except for the 'Set Speed' control, which remains enabled to allow users to configure playback speed before loading a video. Once a video is successfully loaded, the disabled buttons become active, enabling various video operations. Conversely, when the video is closed, the buttons return to their disabled state and will remain so until a new video is loaded..

Visualization

After successfully loading a video, the previously disabled buttons become active, and the progress bar can be dragged to navigate to any point in the video. The Play button starts playback from the current position, while the Pause button pauses the video. When the video is playing, the Play button is disabled to prevent redundant commands; conversely, when paused, the Pause button is disabled. The Close button unloads the current video, reverting the GUI page to its initial blank state. Clicking the Restart button resets the video to the beginning. Playback speed can be adjusted via a drop-down menu, offering options of 0.5×, 1.0×, 1.5×, and 2.0× speeds. Additionally, the video can be fast-forwarded or rewound by 1 or 2 seconds as needed. To clarify the functions of each button, Table 12 provides a detailed description:

Table 12. The functions of buttons on the GUI.

Button	Function description
Load video	Loads a video from a specified path
Play	Plays the video from the current position; playback continues even if the progress bar is manually adjusted
Pause	Pauses the video playback
Close	Closes the loaded video and clears all displayed data
Restart	Restarts the video playback from the beginning
Set speed	Applies the selected playback speed (0.5, 1.0, 1.5, or 2.0) from the drop-down menu
>>1s	Fast-forwards the video for 1 second

>>2s	Fast-forwards the video for 2 seconds
<<1s	Fast back the video for 1 second
<<2s	Fast back the video for 2 seconds

To provide clear operational information to the operator, a data zone has been implemented within the GUI. This zone displays key numerical values such as distance and angle, derived from the processed video frames. For each frame, the system calculates the average of the detected values. For example, if two bounding boxes are identified in a single frame with distances of 0.82 meters and 1.08 meters, the displayed distance will be the average of these values—0.95 meters. To help maintain a safe operational distance, the value labeled "Distance to be travelled" is calculated as the average distance minus 0.5 meters. This adjustment ensures that the underwater robot approaches, but does not collide with, the detected object. Regarding orientation, the "Angle to be rotated" is defined as the negative of the displayed Angle value. This represents the corrective rotation needed to align the robot with the chain link.

It is important to note that the data zone will only function correctly if the output data generated during identification and localization is located in the same directory as the GUI code. If the text document containing the data is overwritten or misplaced, the data visualization will fail. To avoid such issues, it is recommended to save a backup copy of the generated text file to preserve the data for future use. Proper functionality is ensured when both the code and the output files are maintained in the same relative path, as illustrated in Figure 85.

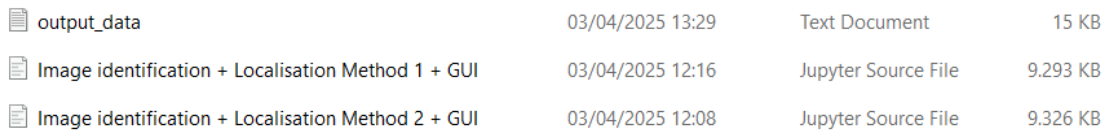


Figure 85. The right relative path between the identification & localization code and the output data

Each time the identification and localization code is executed, both the output video and data files are overwritten unless the file names are manually changed in the code. Therefore, to preserve each run, it is advisable to duplicate and rename the output files accordingly.

Once a video is successfully loaded into the GUI, the interface displays three main zones: Video Zone – Displays the visual output from the processed video; Functional Zone – Contains controls such as playback, speed settings, and navigation; Data Zone – Shows computed distance and angle data in real-time. These three zones together form the complete layout of the GUI. The final GUI for underwater identification and localization is shown in Figure 86.

Figure 86. GUI for underwater identification and localization

D. Image enhancement results

A variety of image enhancement methods are introduced in [142], each producing distinct visual effects. An overview of the enhancement results presented in [142] is shown in Figure 87.

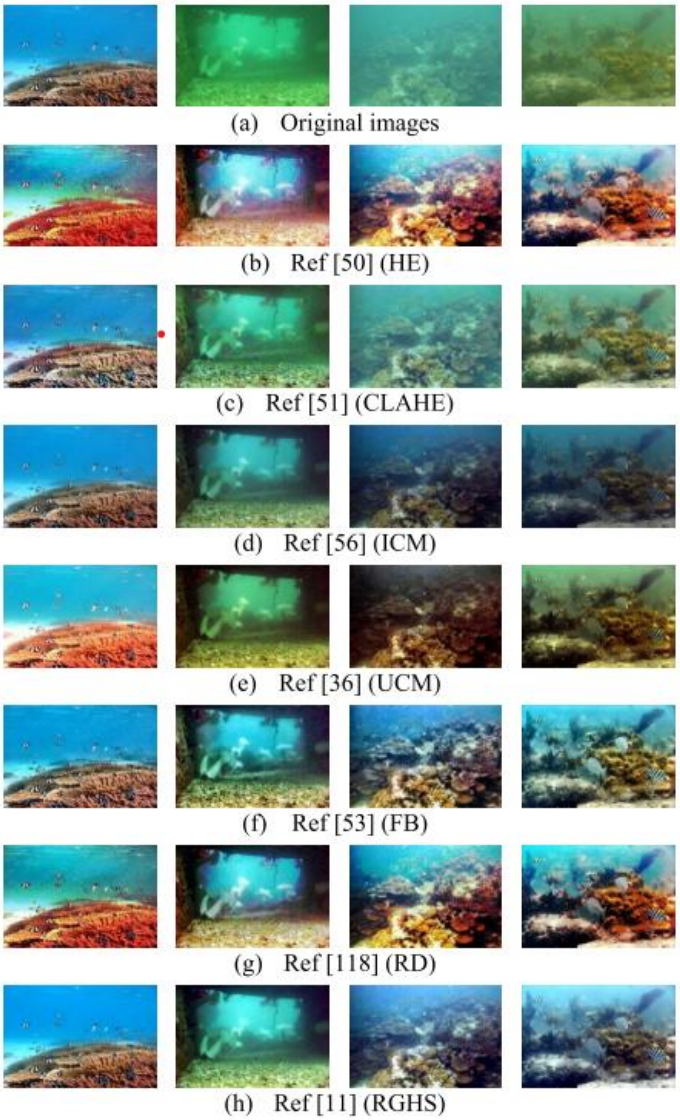


Figure 87. The image enhancement results in [142].

Aside from the results above, some image enhancement results in this assignment which apply methods in [142] are shown in Figure 88.

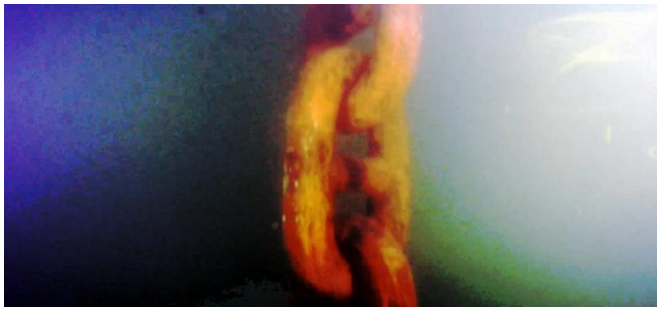
(a) Before enhancement

(b) Enhanced by CLAHE

(c) Enhanced by GC

(d) Enhanced by HE

(e) Enhanced by ICM



(f) Enhanced by RayleighDistribution (RD)

(g) Enhanced by RGHS

Figure 88. Some image enhancement results applied in [142].

Among these enhancement results, Figure (b) CLAHE, (e) ICM, and (g) RGHS produce particularly convincing visual improvements. While Figure (d) HE and (f) RD do enhance the contrast of the target object, they also introduce distortion in the background, making the scene appear unnatural. Of the top-performing methods, (g) RGHS stands out by delivering the highest contrast without compromising image integrity. Therefore, RGHS is selected as the preferred image enhancement technique for this application.

E. Running environmental requirements

Before delving into specific techniques, it is important to introduce the runtime environment used for this project.

All code in this assignment is developed and executed using Python 3.12.4. This version is available on the official Python website. To install it, simply search for "Python 3.12.4" on your preferred search engine, download the installer from the official source, and complete the installation process.

The image enhancement algorithms are implemented using Visual Studio Code (VS Code) as the code editor. You can obtain VS Code by searching for "Visual Studio Code" online and downloading it from its official website. A sample interface of VS Code is shown in Figure 89.

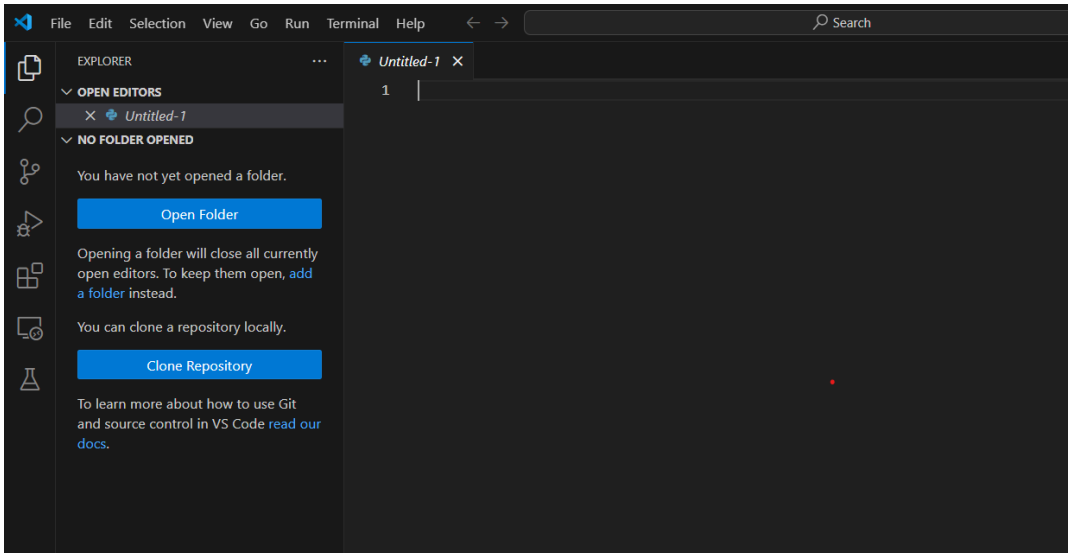


Figure 89. The screen of the VS Code.

The underwater identification and localization component is more complex and involves several stages, including dataset download, model training, validation, and testing. To facilitate these steps, Jupyter Notebooks are used within VS Code.

Jupyter Notebooks allow for cell-by-cell code execution, making it easier to manage and debug individual modules without running the entire script at once. This helps avoid excessive memory usage and potential conflicts between different code segments. To use Jupyter Notebooks in VS Code, search for "Jupyter" in the Extensions tab and install it. An example of a Jupyter Notebook interface in VS Code is shown in Figure 90.

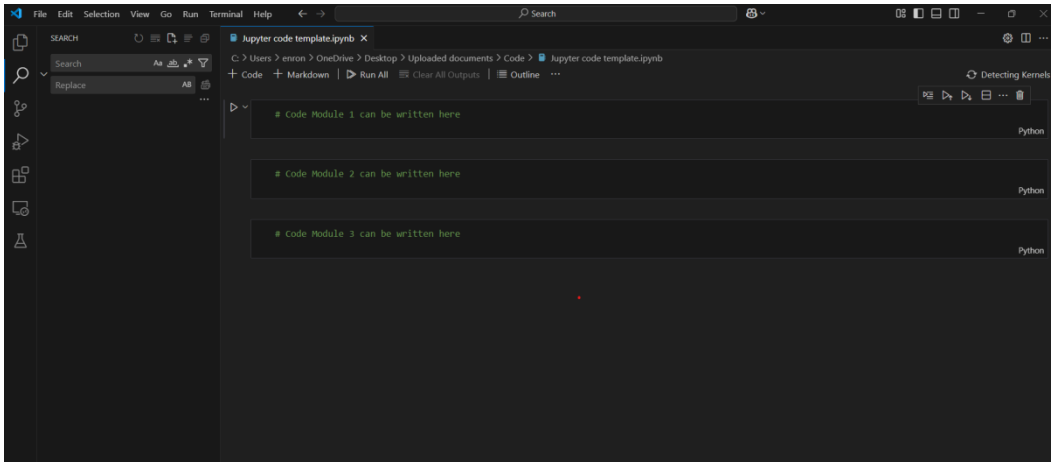


Figure 90. The screen of the Jupyter notebook in VS code

All required libraries for identification and localization are listed in the provided code files. Most of them should be installed automatically. However, if any dependencies are missing, they can be manually installed via the command prompt using the pip package manager. In some cases, the Visual Studio Installer may also be required to support certain packages. A list of required tools is shown in Figure 91.

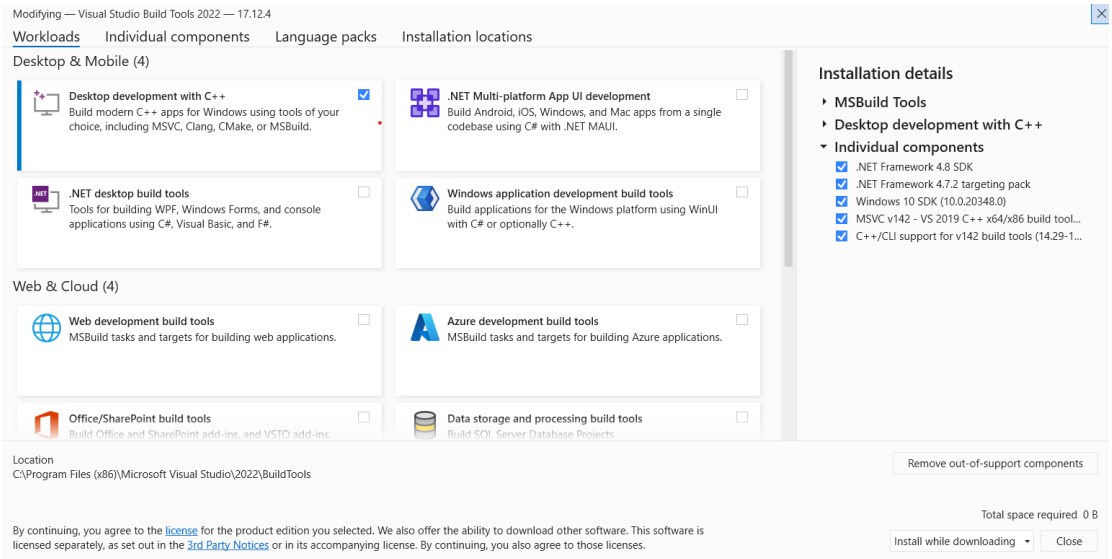


Figure 91. The required tool in Visual Studio Installer for underwater identification and localization.

F. Training dataset for deep learning

A suitable training dataset is essential for developing a robust identification model. Various publicly available datasets exist for underwater object identification, each offering distinct characteristics and use cases. For instance, M. J. Islam et al. introduced the EUVP (Enhancing Underwater Visual Perception) dataset [148], which includes paired and unpaired image samples of varying perceptual quality. This dataset supports supervised training of underwater image enhancement models and is structured to facilitate improvements in visual clarity. An overview of the EUVP dataset resource page is shown in Figure 92.

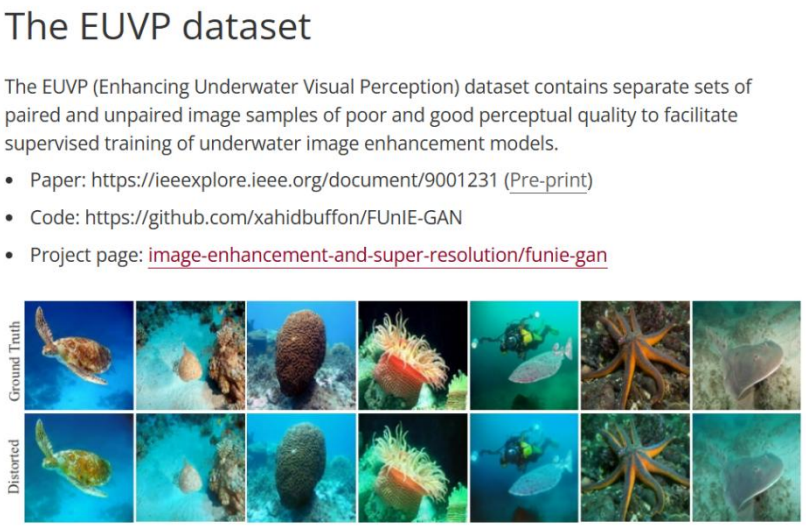


Figure 92. The resources page of EUVP dataset [149].

In another contribution, M. J. Er [150] introduced a collection of underwater datasets such as Fish4Knowledge, LifeCLEF 2014, URPC, UDD, and DUO. These datasets primarily focus on marine organism identification and support a wide range of underwater research applications.

Jian M. et al. [151] also provided a comprehensive summary of available underwater datasets. In addition to marine organism data, they included datasets such as TrashCan, which contains 7,212 images of various underwater objects with challenging visual characteristics. A summary of the datasets discussed by Jian M. et al. [151] is provided in Table 13.

Table 13. The datasets for underwater object detection introduced by Jian, M. et al in [151].

Dataset	Number	Application	Challenge	Year
Brackish	14518	Object/target detection	Poor image quality, too blurry, contains few underwater targets and categories	2019
MUED	8600	Object/target detection Saliency detection	Large dataset is not conducive to model training and validation, and the overall background is too uniform	2019
RUIE	4230	Target detection Target enhancement Target classification	The amount of data is small, and each subset has strong specificity, resulting in poor generalization	2020
UWD	10000	Target detection	Few data categories and uneven sample size	2020
TrashCan	7212	Target detection Target segmentation	The types of underwater targets are complex and difficult to distinguish	2020
UDD	2227	Target detection	Insufficient data volume and uneven distribution of categories, resulting in poor generalization ability due to insufficient samples in certain categories	2021
DUO	7782	Target detection	There are few underwater target categories, and the sample size of each category is uneven	2021

For this assignment, the dataset used is from Roboflow, the world's largest collection of open-source computer vision datasets and APIs [152]. Roboflow offers a user-friendly platform to access and download datasets across a wide range of object categories, making it convenient to locate and obtain the desired training data. The Roboflow homepage is shown in Figure 93.

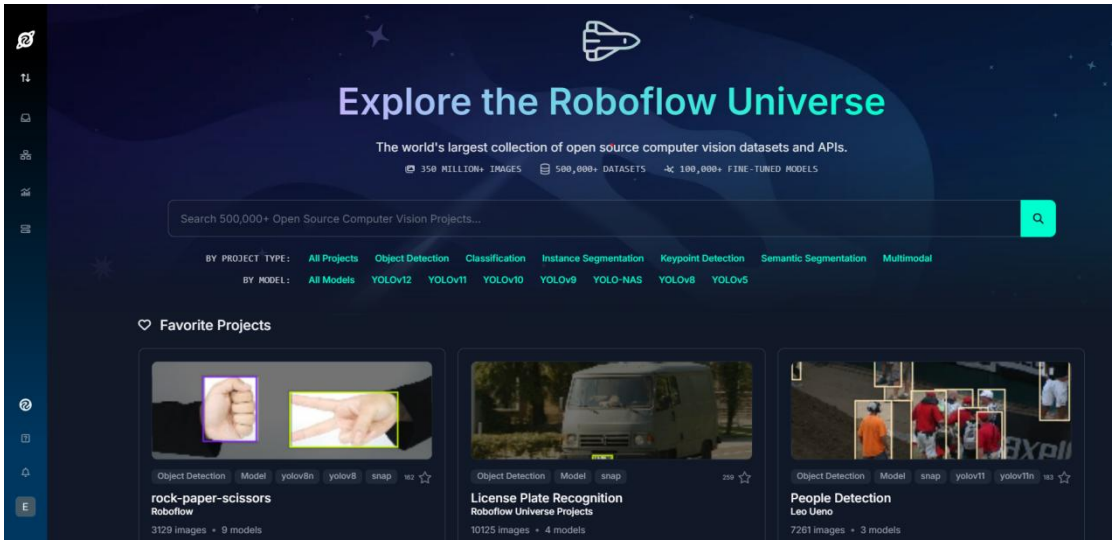


Figure 93. The home page of Roboflow.

The specific dataset applied in this assignment is titled "Chain Link". This dataset comprises a large collection of images featuring chain links in various underwater scenarios, ensuring a diverse and high-quality training source. The dataset includes pre-labeled Regions of Interest (ROIs), which significantly accelerates the training process by providing ready-to-use annotations. An overview of this dataset is shown in Figure 94.

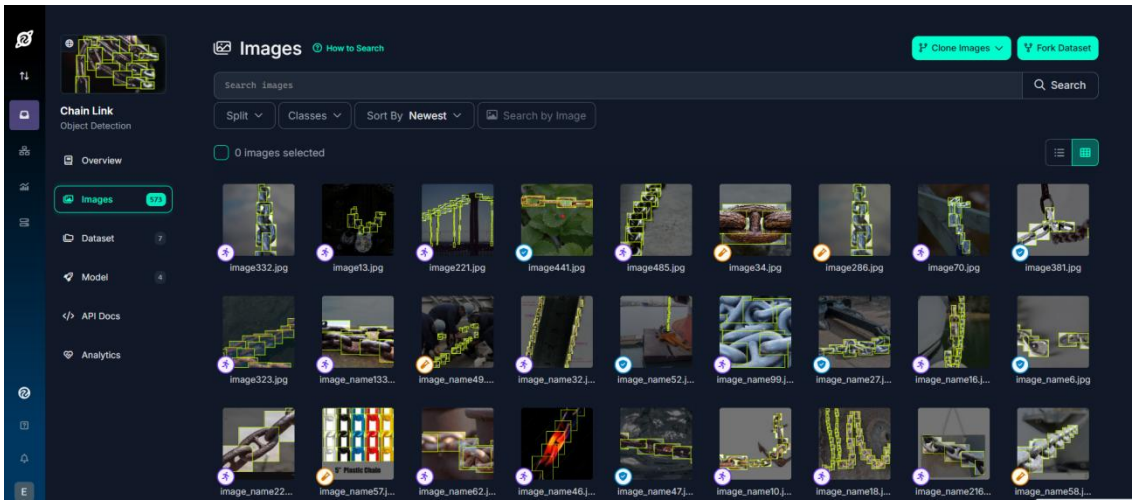


Figure 94. The overview of the 'Chain link' training dataset.

Roboflow supports multiple export formats compatible with popular identification frameworks such as YOLO (You Only Look Once), Detectron2, and EfficientDet-PyTorch. This flexibility allows seamless integration of the dataset with different model architectures. The available export formats are illustrated in Figure 95.

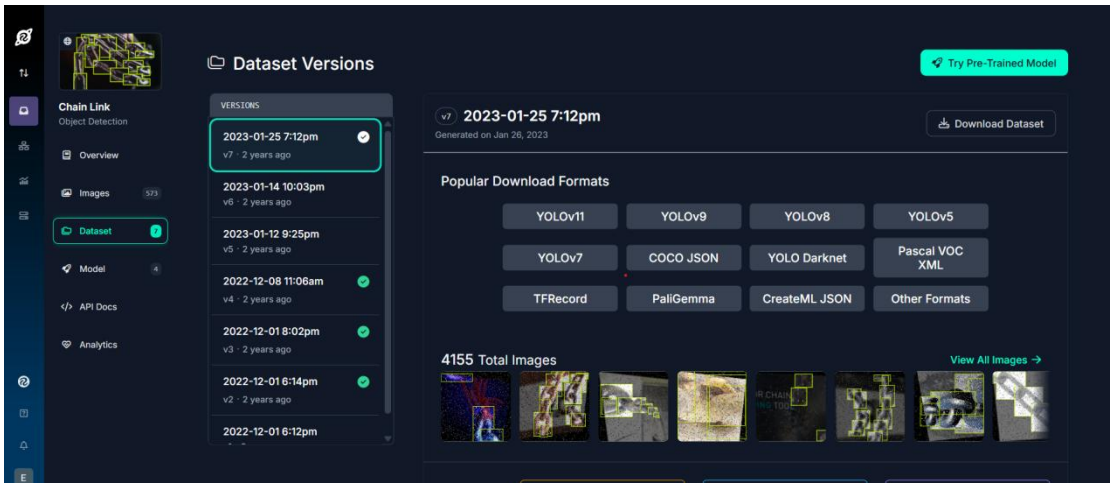


Figure 95. The data formats that are available in Roboflow.

G. Alternative method for localization

The methods introduced in Section 3.3 rely heavily on the use of a Graphics Processing Unit (GPU) for deep learning computations. However, not all laptops or personal computers are equipped with a powerful GPU, limiting accessibility for some users. To address this constraint, an alternative localization method—Method 3, based on depth camera technology—is proposed.

This approach utilizes a depth camera to directly acquire positional data, eliminating the need for computationally intensive deep learning models. The technology is inspired by the principle of human binocular vision, where two cameras are positioned a few centimeters apart. When both sensors capture the same scene, a particular feature will appear at slightly different locations in each image due to the difference in viewpoint. The system calculates this disparity and applies triangulation to estimate the depth of each point in the scene [153]. The basic principle behind this technique is discussed in Section 1.4.1: Preliminary Knowledge – Underwater Localization – Triangulation.

Several depth camera models are available on the market, with two of the most widely used types being Intel RealSense Depth Cameras and Luxonis OAK-D (DepthAI Stereo Cameras). The RealSense D400 series depth camera can realize powerful image processing, offering quality depth for a variety of applications. Its wide field of view is perfect for applications such as robotics, augmented and virtual reality, robotic navigation and object recognition [145]. The most representative models of Intel RealSense D400 series are indicated in Table 14.

Table 14. The most representative models of Intel RealSense D400 series [154, 155].

Model	Description
D435	Developing cutting-edge vision sensing products, combining an Intel module and vision processor together into a small form factor and thus yields a solution ideal for both development and rapid product creation.
D455	The camera integrates an IMU to refine its depth awareness in any situation where the camera moves. This allows improved environmental awareness for robotics and drones. What is more, the depth error is reduced to less than 2% at 4m.
D456	The D456 has an IP65 rated enclosure which is dust tight and protected from projected water, which makes it the best fit for outdoor applications (under extreme environments), such as outdoor robots, automotive infotainment outdoor digital signs and more.

Apart from physical products, algorithms are essential for the acquisition of distance. Intel provides users with depth estimation code source for RealSense D400 series, which is named Intel RealSense SDK 2.0. The code source supports various platforms and programming languages. In Figure 96, the relevant code source is shown.

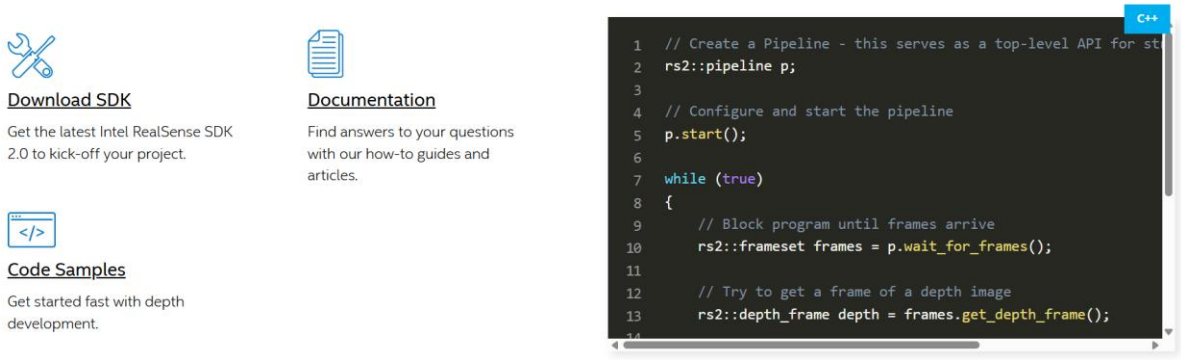


Figure 96. The relevant code source for Intel RealSense D400 series [156].

As the other type of depth camera model, OAK-D is the ultimate camera for robotic vision that perceives the world like a human by combining stereo depth camera and high-resolution color camera with Neural Network inferencing and Computer Vision capabilities. It uses USB-C for both power and USB3 connectivity. The relevant parameters and their values for OAK-D based depth estimation are indicated in Table 15.

Table 15. The relevant parameters and their values for Luxonis OAK-D based depth estimation [157].

Relevant parameters	Values
Base line	75mm
Ideal depth range	70cm - 12m
Power consumption	Up to 5W
Dimensions	110x54.5x33 mm
Weight	115g

Similar to Intel RealSense Depth Camera, the algorithm for the Luxonis OAK-D DepthAI Stereo Camera is also required. The depthai-experiments is an open source code which provides experimental projects that have been done with DepthAI. In Figure 97, the depth estimation of human face is shown.

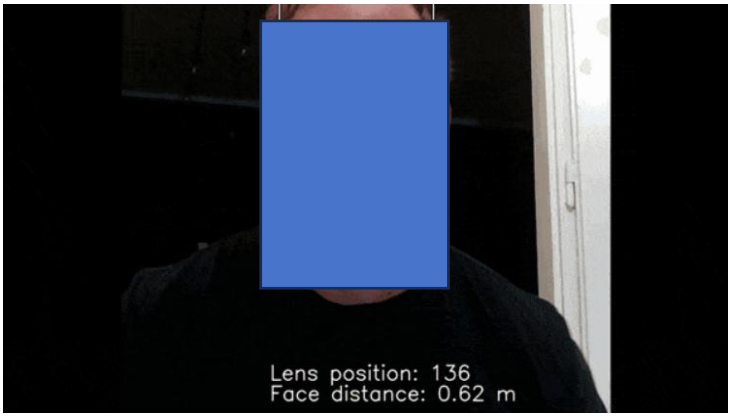


Figure 97. The depth estimation of depthai-experiments algorithm for the Luxonis OAK-D DepthAI Stereo Camera [158].

The depthai-experiments framework has proven effective for estimating the depth of detected objects, making it a promising solution for distance estimation using a depth camera.

H. Alternative models for identification

Two-stage algorithm

In this assignment, the Faster R-CNN X101-FPN model is selected from the available untrained models due to its superior identification accuracy. Although several models are available, those related to Faster R-CNN have already been introduced in Section 3.2.1. This section presents additional common untrained models. The most representative models within the Detectron2 framework are summarized in Table 16 [143].

Table 16. The most typical models of Detectron2.

RetinaNet:

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	model id	download
R50	1x	0.205	0.041	4.1	37.4	190397773	model | metrics
R50	3x	0.205	0.041	4.1	38.7	190397829	model | metrics
R101	3x	0.291	0.054	5.2	40.4	190397697	model | metrics

RPN & Fast R-CNN:

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	prop. AR	model id	download
RPN R50-C4	1x	0.130	0.034	1.5		51.6	137258005	model | metrics
RPN R50-FPN	1x	0.186	0.032	2.7		58.0	137258492	model | metrics
Fast R-CNN R50-FPN	1x	0.140	0.029	2.6	37.8		137635226	model | metrics

For an expanded list of model options, please refer to reference [143]. It is important to note that the algorithms used in Detectron2 are two-stage object detection algorithms. These typically offer higher detection accuracy compared to one-stage algorithms, but this comes at the cost of slower processing speed.

To use a different untrained model within Detectron2, a simple modification in the code configuration is required. The specific line in the code where the model can be changed is highlighted in Figure 98.

```
# Train the identification model
from detectron2.engine import DefaultTrainer

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ("chainlink_train",)
cfg.DATASETS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml") # Let training initialize from model zoo
cfg.SOLVER.IMS_PER_BATCH = 2  # This is the real "batch size" commonly known to deep learning people
cfg.SOLVER.BASE_LR = 0.00025  # pick a good LR
cfg.SOLVER.MAX_ITER = 500     # 300 iterations seems good enough for this toy dataset; you will need to train longer for a practical dataset
cfg.SOLVER.STEPS = []         # do not decay learning rate
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128   # The "RoIHead batch size". 128 is faster, and good enough for this toy dataset (default: 512)
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1  # only has one class (ballon). (see https://detectron2.readthedocs.io/tutorials/datasets.html#update the config for new datasets)
# NOTE: this config means the number of classes, but a few popular unofficial tutorials incorrect uses num_classes+1 here.

cfg.OUTPUT_DIR = "./output"
os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

Figure 98. The place which should be modified to change the model.

One-stage algorithm

In the real-time detection scenario, the processing speed of identification model becomes a dominant factor, therefore, the one-stage algorithm is chosen. Among all one-stage algorithms, the YOLO series are the most popular. Some representative algorithms include: YOLO v5, YOLO v7, YOLO v8, YOLO v9 and YOLO v11.

YOLO v5

YOLO v5 represents an advancement in object detection methodologies, YOLO v5 integrates the anchor-free, objectness-free split head, a feature previously introduced in the YOLO v8 models. This adaptation refines the model's architecture, leading to an improved accuracy-speed tradeoff in object detection tasks. The relationship between mAP (Mean Average Precision) and processing speed of Yolo v5 is shown in Figure 99 (Figure 99-103 are of similar characteristics)

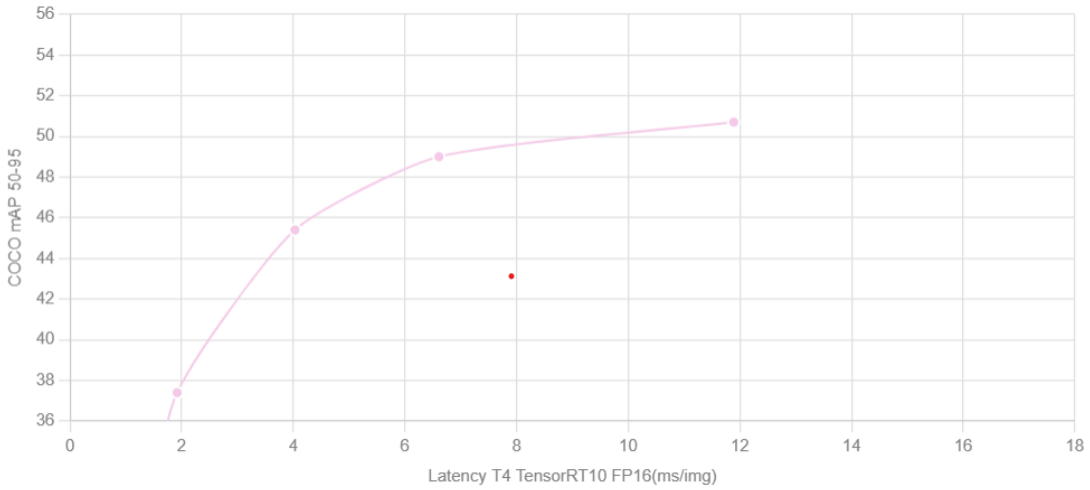


Figure 99. The mAP and processing speed of Yolo v5 [159]. Dots represent different versions of Yolo v5

YOLO v7

YOLO v7 is a state-of-the-art real-time object detector that surpasses all known object detectors in both speed and accuracy in the range from 5 FPS to 160 FPS. It has the highest accuracy (56.8% AP) among all known real-time object detectors with 30

FPS or higher on GPU V100.

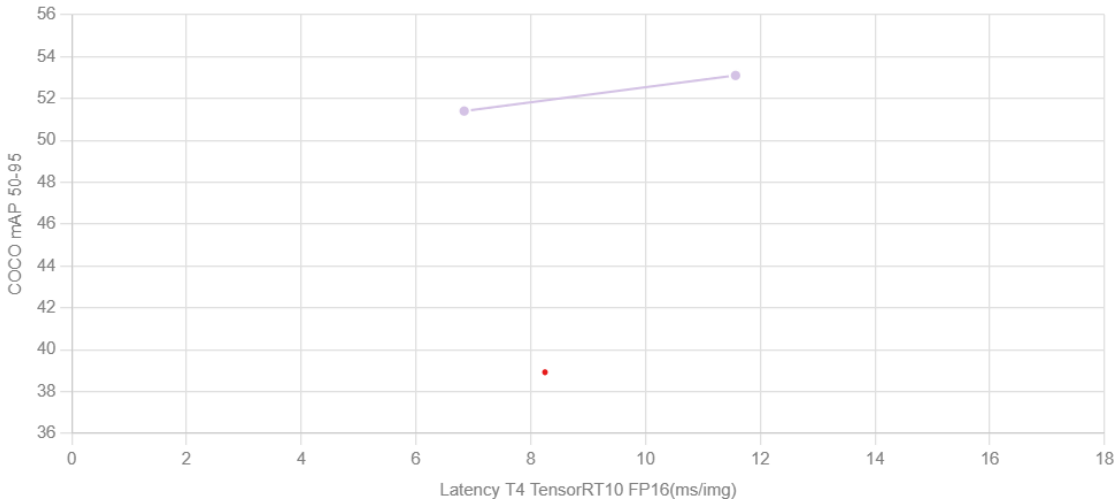


Figure 100. The mAP and processing speed of Yolo v7. Dots represent different versions of Yolo v7

YOLO v8

YOLO v8 offers cutting-edge performance in terms of accuracy and speed. Building upon the advancements of previous YOLO versions, YOLO v8 introduced new features and optimizations that make it an ideal choice for various object detection tasks in a wide range of applications.

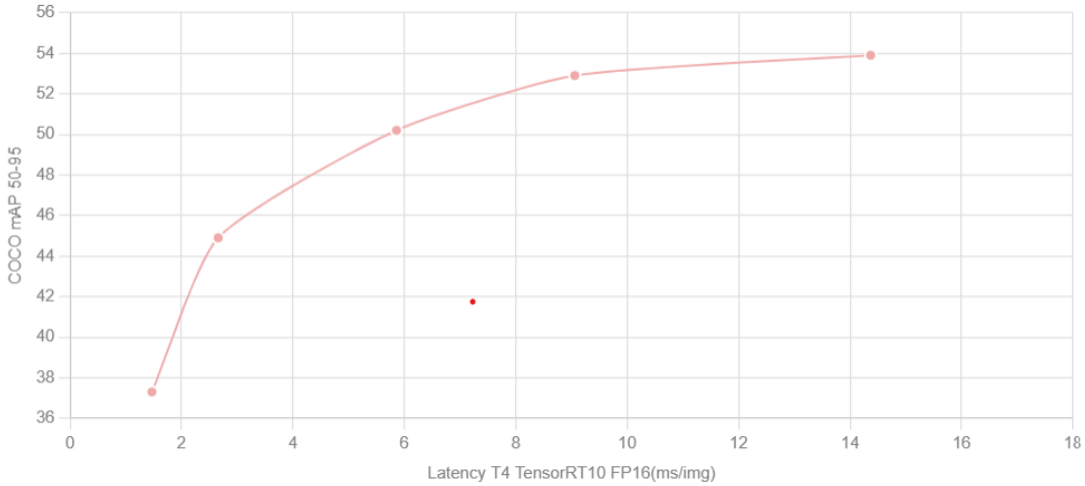


Figure 101. The mAP and processing speed of Yolo v8. Dots represent different versions of Yolo v8

YOLO v9

YOLO v9 marks a significant advancement in real-time object detection, introducing groundbreaking techniques such as Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN). This model demonstrates remarkable improvements in efficiency, accuracy, and adaptability, setting new benchmarks on the MS COCO dataset.

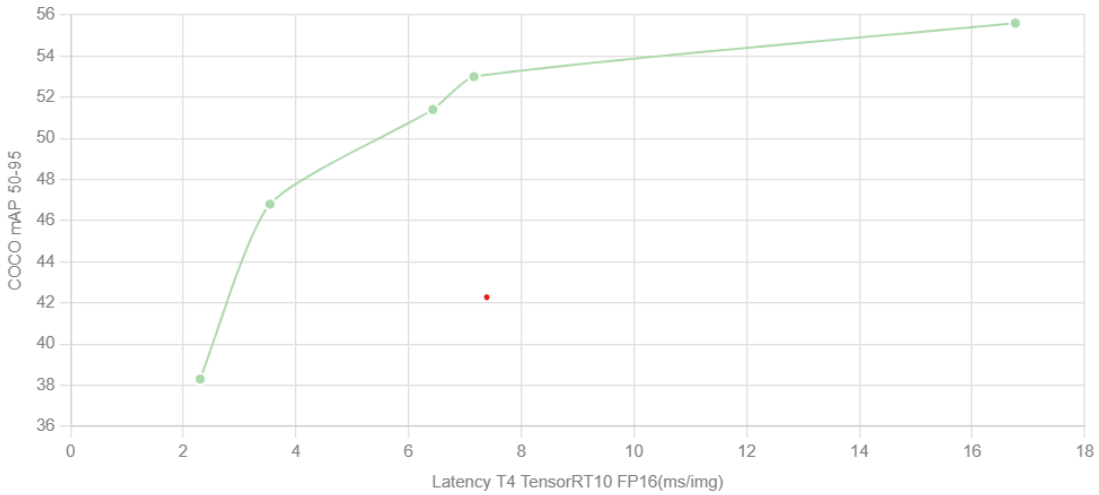


Figure 102. The mAP and processing speed of Yolo v9. Dots represent different versions of Yolo v9

YOLO v11

Yolo v11 is the latest iteration in the Ultralytics YOLO series of real-time object detectors, which further improves accuracy, speed, and efficiency. YOLO v11 introduces significant improvements in architecture and training methods, making it a versatile choice for a wide range of computer vision tasks.

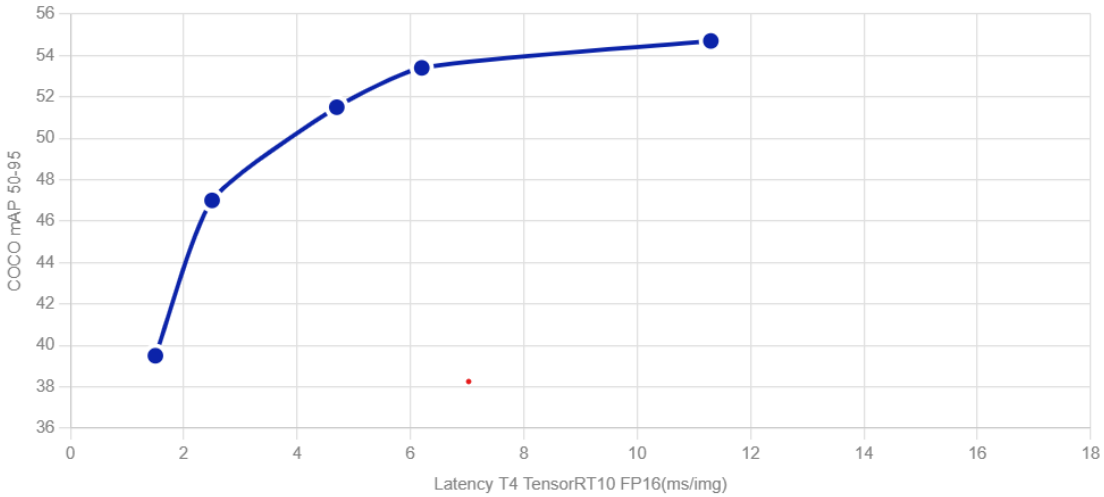


Figure 103. The mAP and processing speed of Yolo v11. Dots represent different versions of Yolo v11

I. Alternative depth estimation algorithm

MiDas

In this assignment, Metric3D was selected as the primary metric depth estimation algorithm due to its strong performance compared to other metric-based methods. However, the processing speed of the algorithms under Metric3D is relatively low, as shown in Table 17.

Table 17. The processing speed of Metric 3D algorithms [160].

Model	Speed
ConvNeXt-Large	10.5 fps
ViT-Small	11.6 fps
ViT-Large	9.5 fps
ViT-giant	5.0 fps

Due to these limitations in processing speed, there is a need for alternative approaches that offer faster performance. Depth estimation methods can generally be classified into two categories: metric depth estimation and relative depth estimation.

One promising alternative to Metric3D is MiDaS, a robust monocular relative depth estimation algorithm. MiDaS was trained on up to 12 diverse datasets, enhancing its generalization capabilities. Unlike metric depth methods, MiDaS omits some of the computationally intensive transformations, which leads to significantly improved processing speeds. An overview of MiDaS's processing performance and advancements is presented in Figure 104.

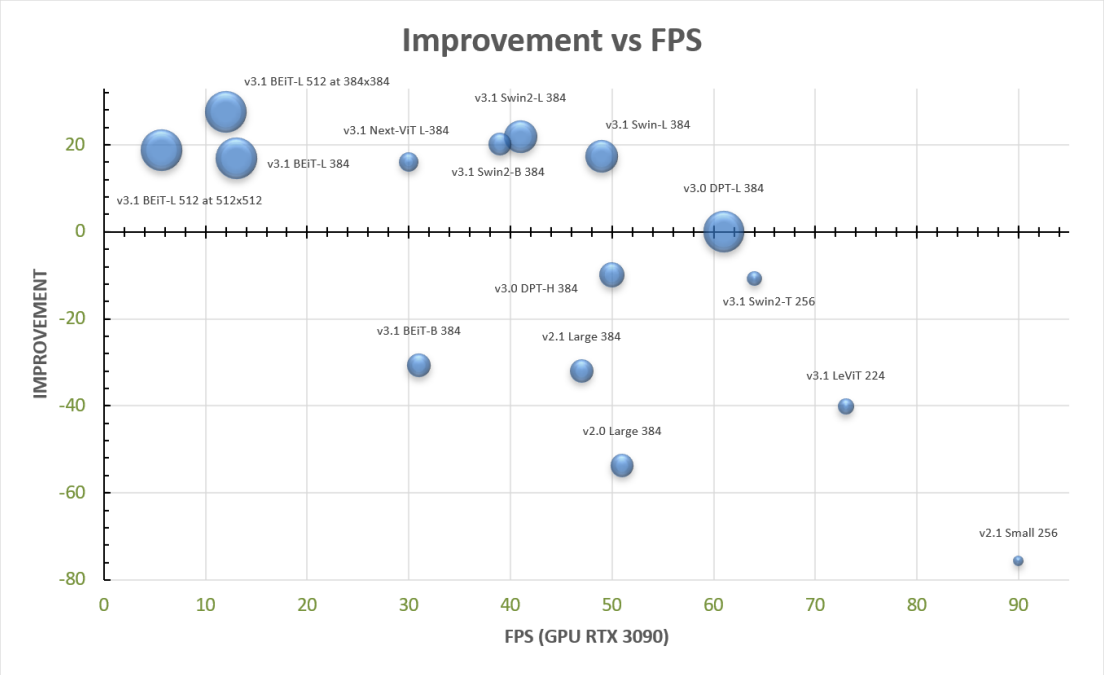
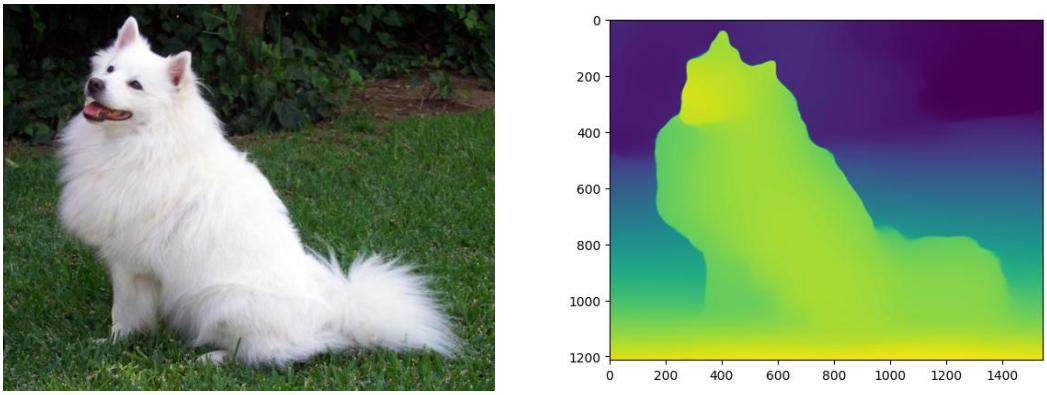


Figure 104. Processing speed and improvement of algorithms in MiDas [161].

Unlike metric depth estimation, MiDaS generates relative depth maps, where each pixel value represents the relative distance of the object from the camera. Although it does not provide absolute measurements in meters, relative depth is often sufficient to distinguish object positions and detect spatial structures. For instance, Figure 105 shows an example of a raw input image and its corresponding predicted depth map generated by MiDaS.



(a) Raw image (b) Predicted depth map

Figure 105. The raw image and predicted depth map processed by MiDas [162].

In the depth map, objects close to the camera (e.g., the dog) are represented by higher pixel values (e.g., around 30), while distant background elements have significantly lower values (e.g., less than 5). This stark contrast indicates that MiDaS can effectively differentiate between foreground and background, making it a viable alternative for fast and practical depth estimation in real-world applications.

Depth Anything v2

Depth Anything V1 is a highly practical solution for robust monocular depth estimation by training on a combination of 1.5M labeled images and 62M+ unlabeled images. Depth Anything V2, which significantly outperforms V1 in fine-grained details and robustness, is a high-performance metric depth estimation model that delivers competitive results comparable to Metric3D. One of its key advantages lies in the significantly smaller size of its pre-trained models, which translates to reduced computational burden. A comparative overview of the model sizes for Depth Anything V2 and Metric3D is provided in Table 18.

Table 18. The comparison between the models of Depth Anything v2 and Metric 3D

Algorithm	Model type	Size
Metric 3D	Metric_depth_vit_small	143M
	Metric_depth_vit_large	1.5G
	Metric_depth_vit_giant2	5.51G
Depth Anything v2	Depth-Anything-V2-Small	24.8M
	Depth-Anything-V2-Base	97.5M
	Depth-Anything-V2-Large	335.3M

In general, larger model sizes demand more computational resources and memory, making them less suitable for real-time applications or resource-constrained environments. As shown in Table 18, Depth Anything V2 models are significantly smaller than those of Metric3D. While Depth Anything V2 offers lower performance compared to Metric3D, the substantial reduction in model size makes it a more practical and efficient choice in scenarios where computational efficiency is prioritized over absolute precision. The inference speeds of various Depth Anything V2 models on different GPUs are presented in Table 19.

Table 19. The processing speed of models in Depth Anything.

Model	Inference Time on V100 (ms)	A100	RTX4090
Depth-Anything-Small	12	8	3
Depth-Anything-Base	13	9	6
Depth-Anything-Large	20	13	12

Figure 106 indicates the depth map of the dog in Figure 105 (a) utilizing Depth Anything V2.

Figure 106. The depth map of the dog in Figure 93 (a) utilizing Depth Anything V2.

As seen in Figure 106, the level of detail and object separation achieved by Depth Anything V2 surpasses that of MiDaS, suggesting superior performance in depth estimation. This makes Depth Anything V2 an excellent choice for applications where high computational efficiency is required.

J. Validation

Validation is a technique in deep learning which aims to evaluate the performance of models during learning. It is done by separating the data set into training and validating sets and then evaluating the performance of the model on the validation sets. It is noticeable that the validation set is quite different from the testing set, the validation set is commonly used in machine learning to evaluate the model's performance while training the model. Nevertheless, the test set is used in evaluating the model's performance on data it has not seen before.

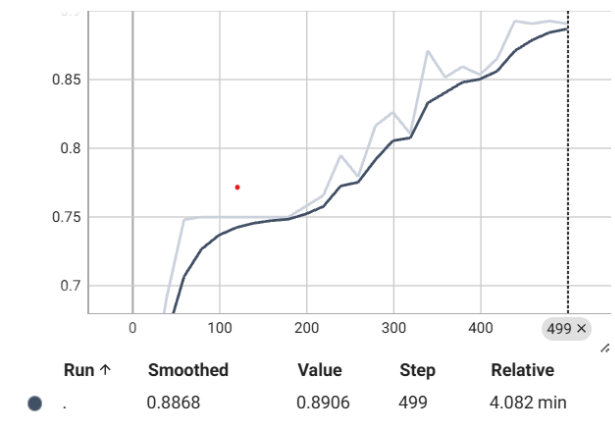
Validation is a necessary step for evaluating the performance before testing. For one, it enables us to detect overfitting and underfitting, which means the suitable training iterations can be found more easily. For another, the parameter tuning for optimal performance can be realized by visualizing the results processed by the trained model

Training iteration exploration

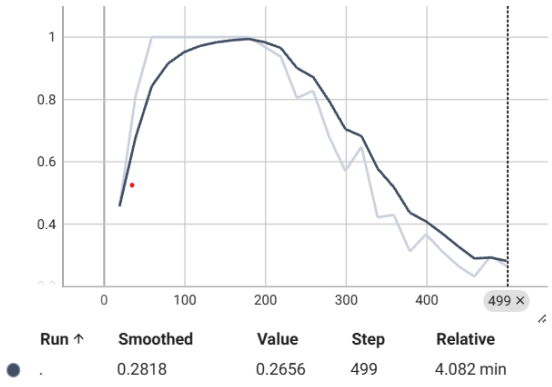
The training iterations largely influence the identification model. If the model is trained with extremely few iterations or inversely, trained with too many iterations, also called underfitting or overfitting, the precision of the model may be both not ideal. In this section, an approximate suitable training iteration is desired to be obtained.

The accuracy and false negative are indexes that are referred to judge the training effect. The higher accuracy is desired to be as high as possible. Accordingly, the false negative is the reverse case. To find the suitable training iterations, the number of iterations is set to be 500 and 600 respectively.

When the training iteration is set to be 500, the accuracy and false negative plots for Faster RCNN X101-FPN are indicated in Figure 107.



(a) cls accuracy (classification accuracy)

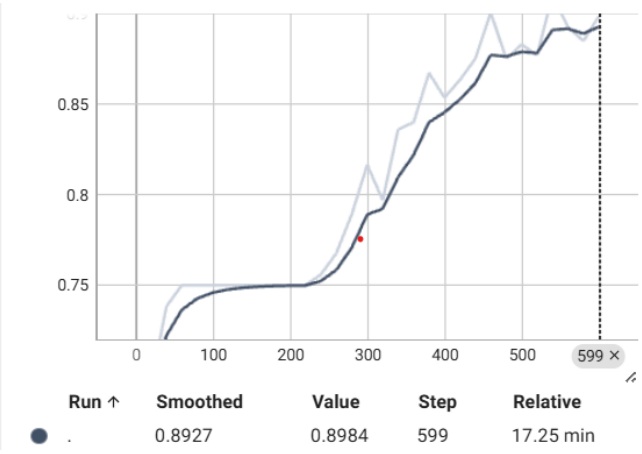


(b) false negative

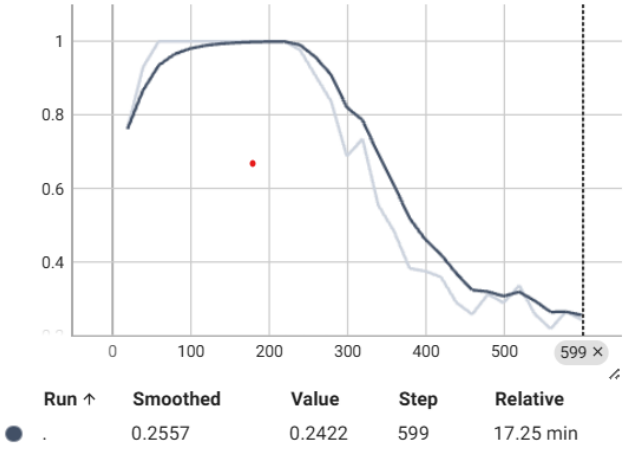
Figure 107. The results of accuracy and false negative plots for Faster RCNN X101-FPN under 500 training iterations.

The results show that the accuracy and false negative plots seem to be reaching a stable state, but it has not reached a very stable state. Therefore, the training iteration is increased to 600.

When the training iteration is selected to be 600, the accuracy and false negative plots for Faster RCNN X101-FPN are indicated in Figure 108.



(a) cls accuracy



(b) false negative

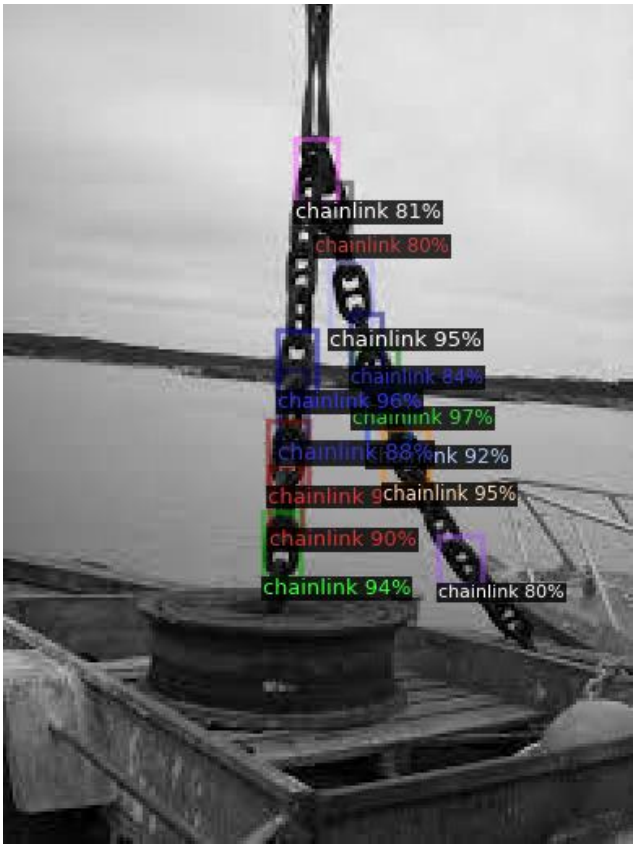
Figure 108. The results of accuracy and false negative plots for Faster RCNN X101-FPN under 600 training iterations.

The cls accuracy is considered classification accuracy. The false negative represents the identification result which wrongly indicates that a particular object does not belong to a specific classification

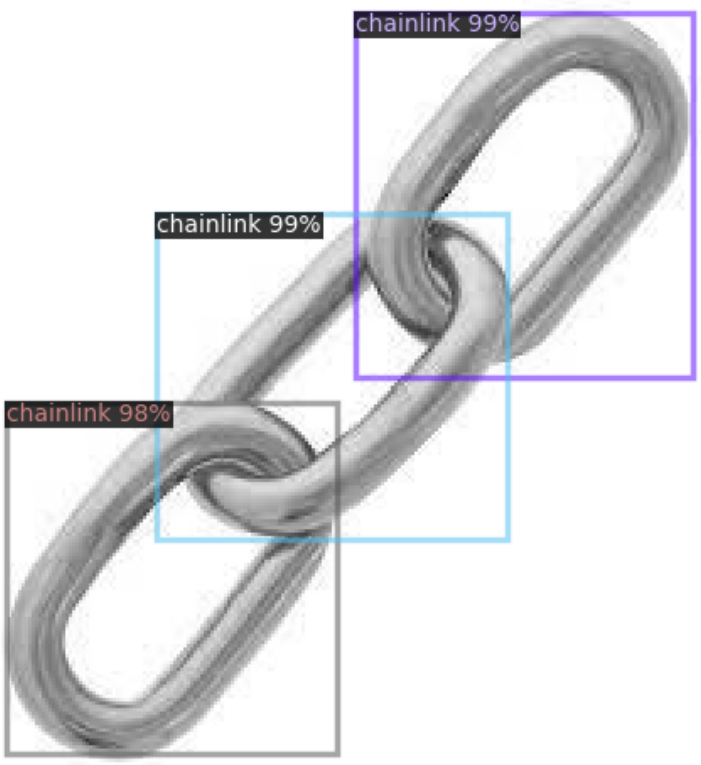
From the results in Figure 108, it can be observed that the model's accuracy approaches 0.9, with the accuracy plot gradually leveling off as the training iteration reaches 600, indicating satisfactory performance. The false negative rate initially increases up to 300 iterations but then drops sharply, and similarly begins to converge around 600 iterations. These trends suggest that 600 is an appropriate choice for the number of training iterations, offering both good and stable performance.

Image visualization

To directly reflect the identification effect of the trained model described above, the identification model carries out the identification process on the images in validation dataset. The validation dataset contains pictures with chain links in different scenarios. The performance of the model can be evaluated by judging the accuracy of the bounding box and its corresponding confidence. Some validation results are indicated in Figure 109.

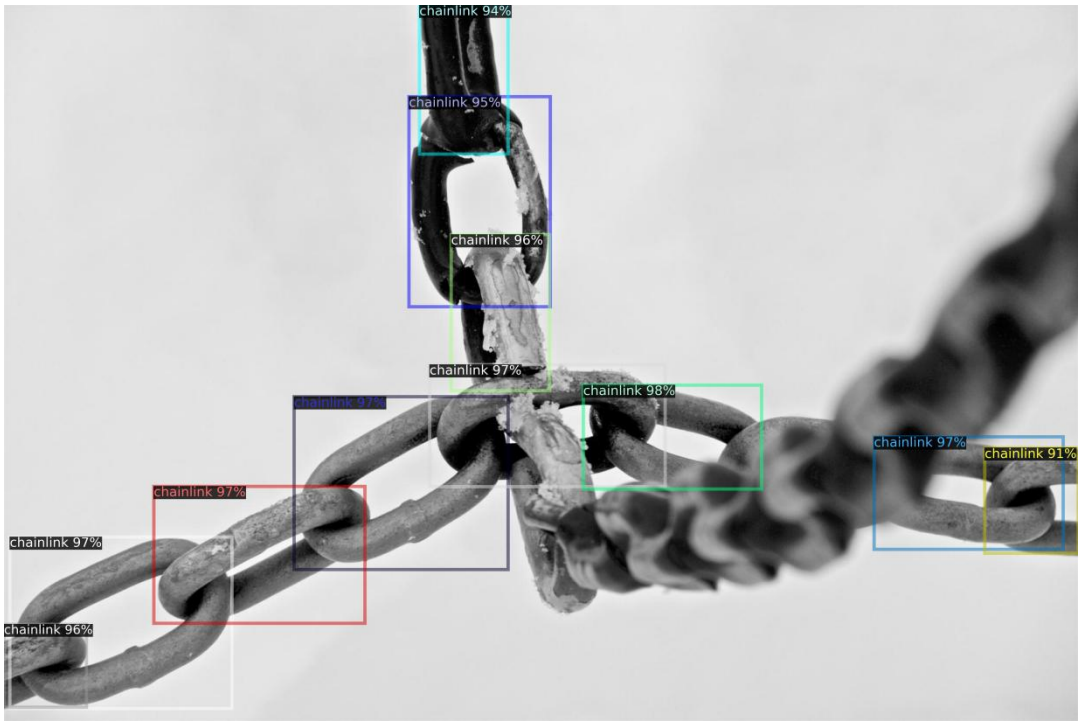


(a) Validation result 1

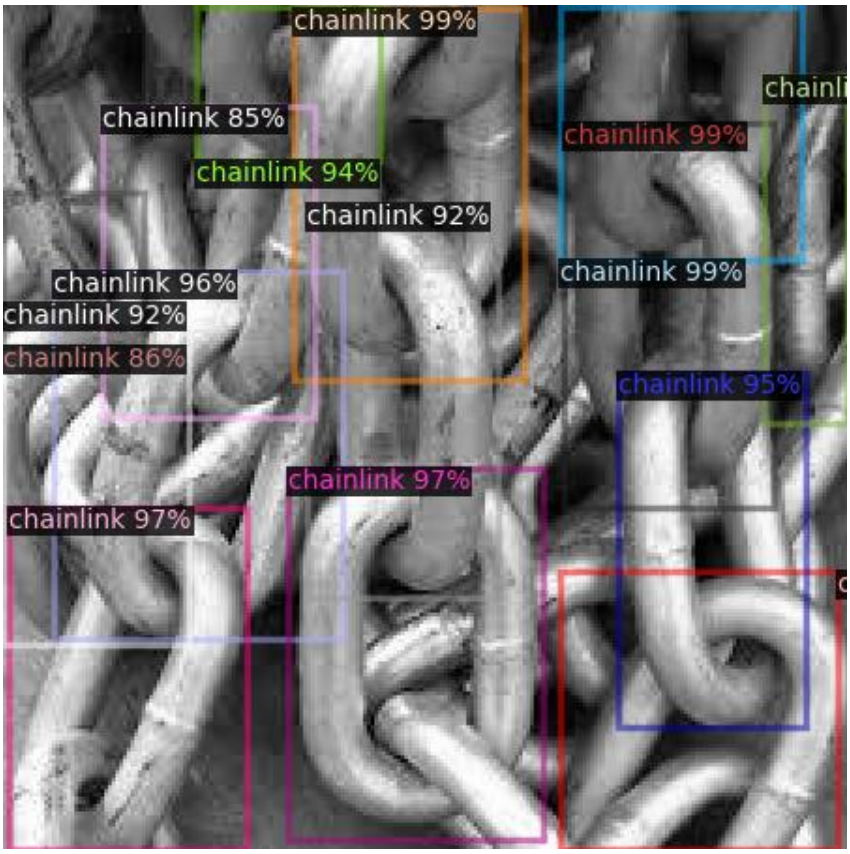


(b) Validation result 2

(c) Validation result 3



(d) Validation result 4



(e) Validation result 5

(f) Validation result 6

Figure 109. Validation results.

From the identification results, it can be found that the accuracy of the bounding box is high, which exceeds 90% most of the time, indicating that the validation result is satisfactory.

K. Underwater operational scenario videos

Before conducting the main experiments, the localization and identification algorithms, along with the graphical user interface (GUI), were initially tested using videos depicting underwater operational scenarios. To assist researchers who may require similar video resources, a list of relevant underwater operational scenario videos is provided in Table 20.

Table 20. The relevant resources for underwater operational scenario videos.

No.	Name	Duration	Website
1	Self-cut Video 1	0:10	https://youtu.be/evFw_78mySM
2	Self-cut Video 2	0:08	https://youtu.be/NB2cNA1ZnZQ
3	Underwater inspection chain of buoy	1:57	https://www.youtube.com/watch?v=hOIS0XcepRM
4	Underwater inspection and survey Mooring Chain & Anchor using ROV.	4:05	https://www.youtube.com/watch?v=yEkduYOstjg
5	ROV Services	1:12	https://www.facebook.com/watch/?v=2025153794187458
6	Holdfast Marine Solutions Ltd. Mooring Inspection	13:10	https://www.youtube.com/watch?v=_Vx1Zn2i_Hc

The Self-cut Videos (1 and 2) were extracted from longer online videos. These original videos contained substantial irrelevant content, so only the high-quality segments clearly displaying chain links were selected and trimmed to create concise, focused clips.

The remaining videos (3–6) are directly related to underwater operations and inspections. As these videos are already highly relevant, they are presented entirely without editing.

Bibliography

[1] King, P., Anstey, B., & Vardy, A. (2017). Sonar image registration for localization of an underwater vehicle.
[2] Xanthidis, M., Joshi, B., Roznere, M., Wang, W., Burgdorfer, N., Li, A. Q., ... & Rekleitis, I. (2022, September). Towards mapping of underwater structures by a team of autonomous underwater vehicles. In The International Symposium of Robotics Research (pp. 170-185). Cham: Springer Nature Switzerland.
[3] Wang, J., Bai, S., & Englot, B. (2017, May). Underwater localization and 3D mapping of submerged structures with a single-beam scanning sonar. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4898-4905). IEEE.
[4] Chen, L., Huang, Y., Dong, J., Xu, Q., Kwong, S., Lu, H., ... & Li, C. (2024). Underwater Object Detection in the Era of Artificial Intelligence: Current, Challenge, and Future. arXiv preprint arXiv:2410.05577.
[5] Fayaz, S., Parah, S. A., & Qureshi, G. J. (2022). Underwater object detection: architectures and algorithms–a comprehensive review. Multimedia Tools and Applications, 81(15), 20871-20916.
[6] Yu, Y., Zhao, J., Gong, Q., Huang, C., Zheng, G., & Ma, J. (2021). Real-time underwater maritime object detection in side-scan sonar images based on transformer-YOLOv5. Remote Sensing, 13(18), 3555.
[7] BlueRobotics. Ping360 Scanning Imaging Sonar. From: https://bluerobotics.com/store/sonars/imaging-sonars/ping360-sonar-r1-rp/
[8] Li, X., & Shao, G. (2014). Object-based land-cover mapping with high resolution aerial photography at a county scale in midwestern USA. Remote Sensing, 6(11), 11372-11390.
[9] Fensham, R. J., & Fairfax, R. J. (2002). Aerial photography for assessing vegetation change: a review of applications and the relevance of findings for Australian vegetation history. Australian Journal of Botany, 50(4), 415-429.
[10] Wang, M., Zhang, K., Wei, H., Chen, W., & Zhao, T. (2024). Underwater image quality optimization: Researches, challenges, and future trends. Image and Vision Computing, 104995.
[11] Chiang, J. Y., Chen, Y. C., & Chen, Y. F. (2011, August). Underwater image enhancement: using wavelength compensation and image dehazing (WCID). In International conference on advanced concepts for intelligent vision systems (pp. 372-383). Berlin, Heidelberg: Springer Berlin Heidelberg.
[12] BlueRobotics. Low-Light HD USB Camera. From: https://bluerobotics.com/store/sensors-cameras/cameras/cam-usb-low-light-r1/
[13] www.scantips.com. Math of Field of View (FOV) for a Camera and Lens. https://www.scantips.com/lights/fieldofviewmath.html
[14] Poldrack, R. A. (2007). Region of interest analysis for fMRI. Social cognitive and affective neuroscience, 2(1), 67-70.
[15] Huang, C., Liu, Q., & Yu, S. (2011). Regions of interest extraction from color image based on visual saliency. The Journal of Supercomputing, 58, 20-33.
[16] Zhang, J., Yoo, C. W., & Ha, S. W. (2007, August). ROI based natural image retrieval using color and texture feature. In Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007) (Vol. 4, pp. 740-744). IEEE.
[17] Wang, Z., Liu, G., & Yang, Y. (2013). A new ROI based image retrieval system using an auxiliary Gaussian weighting scheme. Multimedia tools and applications, 67,

549-569.
[18] Pollicelli, D., Coscarella, M., & Delrieux, C. (2020). RoI detection and segmentation algorithms for marine mammals photo-identification. Ecological Informatics, 56, 101038.
[19] Xu, S., Zhang, M., Song, W., Mei, H., He, Q., & Liotta, A. (2023). A systematic review and analysis of deep learning-based underwater object detection. Neurocomputing, 527, 204-232.
[20] N. Wang, Y. Zhou, F. Han, H. Zhu, J. Yao, UWGAN: Underwater GAN for Realworld Underwater Color Restoration and Dehazing, (2019). https://doi.org/10.48550/ARXIV.1912.10269.
[21] M.J. Islam, Y. Xia, J. Sattar, Fast underwater image enhancement for improved visual perception, IEEE Rob. Autom. Lett. 5 (2020) 3227–3234, https://doi.org/10.1109/LRA.2020.2974710.
[22] X. Liu, Z. Gao, B.M. Chen, MLFcGAN: multilevel feature fusion-based conditional GAN for underwater image color correction, IEEE Geosci. Remote Sens. Lett. 17 (2020) 1488–1492, https://doi.org/10.1109/LGRS.2019.2950056.
[23] A. Naik, A. Swarnakar, K. Mittal, Shallow-UWnet : Compressed Model for Underwater Image Enhancement, (2021).
https://doi.org/10.48550/ARXIV.2101.02073.
[24] Z. Ma, C. Oh, A Wavelet-Based Dual-Stream Network for Underwater Image Enhancement, in: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022: pp. 2769–2773.
https://doi.org/10.1109/ICASSP43922.2022.9747781.
[25] O. Kupyn, T. Martyniuk, J. Wu, Z. Wang, DeblurGAN-v2: deblurring (ordersof-magnitude) faster and better, IEEE/CVF International Conference on Computer Vision (ICCV) 2019 (2019) 8877–8886, https://doi.org/10.1109/ICCV.2019.00897.
[26] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.
[27] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence, 44(7), 3523-3542.
[28] Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., ... & Gao, M. (2023). Techniques and challenges of image segmentation: A review. Electronics, 12(5), 1199.
[29] Peng, Y. T., Lin, Y. C., Peng, W. Y., & Liu, C. Y. (2024). Blurriness-Guided Underwater Salient Object Detection and Data Augmentation. IEEE Journal of Oceanic Engineering.
[30] Jian, M., Qi, Q., Dong, J., Yin, Y., & Lam, K. M. (2018). Integrating QDWD with pattern distinctness and local contrast for underwater saliency detection. Journal of visual communication and image representation, 53, 31-41.
[31] Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
[32] WIKIPEDIA. Support vector machine. From:
https://en.wikipedia.org/wiki/Support_vector_machine
[33] Pal, M. (2005). Random forest classifier for remote sensing classification. International journal of remote sensing, 26(1), 217-222.
[34] Liu, Y., Wang, Y., & Zhang, J. (2012). New machine learning algorithm: Random forest. In Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3 (pp. 246-252).

Springer Berlin Heidelberg.
[35] Zhang, S., Li, X., Zong, M., Zhu, X., & Cheng, D. (2017). Learning k for knn classification. ACM Transactions on Intelligent Systems and Technology (TIST), 8(3), 1-19.
[36] Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach in classification. In On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings (pp. 986-996). Springer Berlin Heidelberg.
[37] IBM. KNN algorithm. https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20(KNN,used%20in%20machine%20learning%20today.
[38] Celebi, M. E., & Aydin, K. (Eds.). (2016). Unsupervised learning algorithms (Vol. 9, p. 103). Cham: Springer.
[39] Builtin. Gaussian Mixture Model Explained. From: https://builtin.com/articles/gaussian-mixture-model
[40] Zhou, L., Zhang, C., Liu, F., Qiu, Z., & He, Y. (2019). Application of deep learning in food: a review. Comprehensive reviews in food science and food safety, 18(6), 1793-1811.
[41] Taheri, R., Ahmed, H., & Arslan, E. (2023). Deep learning for the security of software-defined networks: a review. Cluster Computing, 26(5), 3089-3112.
[42] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019.
[43] Fayaz, S., Parah, S. A., & Qureshi, G. J. (2022). Underwater object detection: architectures and algorithms–a comprehensive review. Multimedia Tools and Applications, 81(15), 20871-20916.
[44] Hussain, M. (2023). YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. Machines, 11(7), 677.
[45] Zeng, N., Wu, P., Wang, Z., Li, H., Liu, W., & Liu, X. (2022). A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. IEEE Transactions on Instrumentation and Measurement, 71, 1-14.
[46] Er, M. J., Chen, J., Zhang, Y., & Gao, W. (2023). Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review. Sensors, 23(4), 1990.
[47] Zhang, M., Xu, S., Song, W., He, Q., & Wei, Q. (2021). Lightweight underwater object detection based on yolo v4 and multi-scale attentional feature fusion. Remote Sensing, 13(22), 4706.
[48] Chen, G., Wang, H., Chen, K., Li, Z., Song, Z., Liu, Y., ... & Knoll, A. (2020). A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal. IEEE Transactions on systems, man, and cybernetics: systems, 52(2), 936-953.
[49] Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. Array, 16, 100258.
[50] Medium. How Data Augmentation Increases Accuracy of your model. From: https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722
[51] Chen, X., Yu, J., Kong, S., Wu, Z., Fang, X., & Wen, L. (2019). Towards real-

time advancement of underwater visual quality with GAN. IEEE Transactions on Industrial Electronics, 66(12), 9350-9359.
[52] Mourikis, A. I., & Roumeliotis, S. I. (2007, April). A multi-state constraint Kalman filter for vision-aided inertial navigation. In Proceedings 2007 IEEE international conference on robotics and automation (pp. 3565-3572). IEEE.
[53] Bahnam, S. A., De Wagter, C., & De Croon, G. C. H. E. (2024). Improving the computational efficiency of ROVIO. Unmanned Systems, 12(03), 589-598.
[54] Ramezani, M., Acharya, D., Gu, F., & Khoshelham, K. (2017). Indoor positioning by visual-inertial odometry. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4, 371-376.
[55] Bourke, P. (1996). Cross correlation. Cross Correlation", Auto Correlation—2D Pattern Identification, 596.
[56] Costa, L. D. F. (2021). Comparing cross correlation-based similarities. arXiv preprint arXiv:2111.08513.
[57] Sun, H., Du, H., Li, M., & He, X. (2021). Study on ray-tracing-based 3D reconstruction method for underwater measurement in glass-flume experiments. Measurement, 174, 108971.
[58] Alvarez, I., Enguita, J. M., Frade, M., Marina, J., & Ojea, G. (2009). On-line metrology with conoscopic holography: beyond triangulation. Sensors, 9(9), 7021-7037.
[59]
[60] Ohki, Makoto, Michael E. Zervakis, and Anastasios N. Venetsanopoulos. "3-D digital filters." Control and Dynamic Systems. Vol. 69. Academic Press, 1995. 49-88.
[61] Brinkmann, R. (2008). The art and science of digital compositing: Techniques for visual effects, animation and motion graphics. Morgan Kaufmann.
[62] Slideshare. Image Restoration and Reconstruction (Noise Removal). From: https://www.slideshare.net/slideshow/chapter-5-78558023/78558023
[63] Guan, R. P., Ristic, B., Wang, L., & Evans, R. (2018). Monte Carlo localization of a mobile robot using a Doppler–Azimuth radar. Automatica, 97, 161-166.
[64] Menegatti, E., Zoccarato, M., Pagello, E., & Ishiguro, H. (2004). Image-based Monte Carlo localization with omnidirectional images. Robotics and Autonomous Systems, 48(1), 17-30.
[65] Mathworks. Localize TurtleBot Using Monte Carlo Localization Algorithm. From: https://nl.mathworks.com/help/nav/ug/localize-turtlebot-using-monte-carlo-localization.html?searchHighlight=AMCL&s_tid=srchtitle_support_results_1_AMCL
[66] Deng, L., Liu, Z., Zhang, T., & Yan, Z. (2023). Study of visual SLAM methods in minimally invasive surgery. Mathematical Biosciences and Engineering, 20(3), 4388-4402.
[67] Biswas, S., & Hazra, R. (2018). Robust edge detection based on Modified Moore-Neighbor. Optik, 168, 931-943.
[68] Younsi, M., Yesli, S., & Diaf, M. (2024). Depth-based human action recognition using histogram of templates. Multimedia Tools and Applications, 83(14), 40415-40449.
[69] Mousavi, V., Varshosaz, M., Remondino, F., Pirasteh, S., & Li, J. (2022). A two-step descriptor-based keypoint filtering algorithm for robust image matching. IEEE Transactions on Geoscience and Remote Sensing, 60, 1-21.
[70] Han, K., Lee, Y., & Choi, H. (2012). Developing an efficient landmark for autonomous docking tasks of underwater robots. 2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 357-361.

https://doi.org/10.1109/URAI.2012.6463016.
[71] Negre, A., Pradalier, C., & Dunbabin, M. (2008). Robust vision-based underwater homing using self-similar landmarks. Journal of Field Robotics, 25(6-7), 360-377.
[72] Du, C. J., & Sun, D. W. (2004). Recent developments in the applications of image processing techniques for food quality evaluation. Trends in food science & technology, 15(5), 230-249.
[73] Dougherty, G. (2009). Digital image processing for medical applications. Cambridge University Press.
[74] Reggiannini, M., & Moroni, D. (2020). The use of saliency in underwater computer vision: A review. Remote Sensing, 13(1), 22.
[75] Himri, K., Ridao, P., & Gracias, N. (2021). Underwater object recognition using point-features, bayesian estimation and semantic information. Sensors, 21(5), 1807.
[76] Wang, N., Zheng, H., & Zheng, B. (2017). Underwater image restoration via maximum attenuation identification. IEEE Access, 5, 18941-18952.
[77] Liang, Z., Wang, Y., Ding, X., Mi, Z., & Fu, X. (2021). Single underwater image enhancement by attenuation map guided color correction and detail preserved dehazing. Neurocomputing, 425, 160-172.
[78] Cheng, H. D., Jiang, X. H., Sun, Y., & Wang, J. (2001). Color image segmentation: advances and prospects. Pattern recognition, 34(12), 2259-2281.
[79] Bosse, S., & Kasundra, P. (2022). Robust Underwater Image Classification Using Image Segmentation, CNN, and Dynamic ROI Approximation. Engineering Proceedings, 27(1), 82.
[80] Chen, Z., Zhang, Z., Dai, F., Bu, Y., & Wang, H. (2017). Monocular vision-based underwater object detection. Sensors, 17(8), 1784.
[81] Jian, M., Liu, X., Luo, H., Lu, X., Yu, H., & Dong, J. (2021). Underwater image processing and analysis: A review. Signal Processing: Image Communication, 91, 116088.
[82] Luo, X., Chen, L., Zhou, H., & Cao, H. (2023). A survey of underwater acoustic target recognition methods based on machine learning. Journal of Marine Science and Engineering, 11(2), 384.
[83] Raj, M. V., & Murugan, S. S. (2019, December). Underwater image classification using machine learning technique. In 2019 International Symposium on Ocean Technology (SYMPOL) (pp. 166-173). IEEE.
[84] Song, G., Guo, X., Wang, W., Ren, Q., Li, J., & Ma, L. (2021). A machine learning-based underwater noise classification method. Applied Acoustics, 184, 108333.
[85] Villon, S., Chaumont, M., Subsol, G., Villéger, S., Claverie, T., & Mouillot, D. (2016, October). Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between Deep Learning and HOG+ SVM methods. In International Conference on Advanced Concepts for Intelligent Vision Systems (pp. 160-171). Cham: Springer International Publishing.
[86] Jian, M., Yang, N., Tao, C., Zhi, H., & Luo, H. (2024). Underwater object detection and datasets: a survey. Intelligent Marine Technology and Systems, 2(1), 9.
[87] Langner, F., Knauer, C., Jans, W., & Ebert, A. (2009, May). Side scan sonar image resolution and automatic object detection, classification and identification. In OCEANS 2009-EUROPE (pp. 1-8). IEEE.
[88] Dos Santos, M., Ribeiro, P. O., Núñez, P., Drews-Jr, P., & Botelho, S. (2017). Object classification in semi structured enviroment using forward-looking sonar. Sensors, 17(10), 2235.

[89] Luo, X., Chen, L., Zhou, H., & Cao, H. (2023). A survey of underwater acoustic target recognition methods based on machine learning. Journal of Marine Science and Engineering, 11(2), 384.
[90] Ge, H., Dai, Y., Zhu, Z., & Liu, R. (2022). A deep learning model applied to optical image target detection and recognition for the identification of underwater biostructures. Machines, 10 (9), 809.
[91] Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. Mechanical Systems and Signal Processing, 107, 241-265.
[92] Mamoshina, P., Vieira, A., Putin, E., & Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. Molecular pharmaceutics, 13(5), 1445-1454.
[93] Choudhary, K., DeCost, B., Chen, C., Jain, A., Tavazza, F., Cohn, R., ... & Wolverton, C. (2022). Recent advances and applications of deep learning methods in materials science. npj Computational Materials, 8(1), 59.
[94] Ge, H., Dai, Y., Zhu, Z., & Liu, R. (2022). A deep learning model applied to optical image target detection and recognition for the identification of underwater biostructures. Machines, 10 (9), 809.
[95] Wang, N., Wang, Y., & Er, M. J. (2022). Review on deep learning techniques for marine object recognition: Architectures and algorithms. Control Engineering Practice, 118, 104458.
[96] Madhan, E. S., Kannan, K. S., Rani, P. S., Rani, J. V., & Anguraj, D. K. (2021). A distributed submerged object detection and classification enhancement with deep learning. Distrib. Parallel Databases, 1-17.
[97] Mittal, S., Srivastava, S., & Jayanth, J. P. (2022). A survey of deep learning techniques for underwater image classification. IEEE Transactions on Neural Networks and Learning Systems, 34(10), 6968-6982.
[98] Li, Y., Lu, H., Li, J., Li, X., Li, Y., & Serikawa, S. (2016). Underwater image de-scattering and classification by deep neural network. Computers & Electrical Engineering, 54, 68-77.
[99] Xu, S., Zhang, M., Song, W., Mei, H., He, Q., & Liotta, A. (2023). A systematic review and analysis of deep learning-based underwater object detection. Neurocomputing, 527, 204-232.
[100] Er, M. J., Chen, J., Zhang, Y., & Gao, W. (2023). Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review. Sensors, 23(4), 1990.
[101] Guntha, P., & Beaulah, P. M. R. (2024, April). A Comprehensive Review on Underwater Object Detection Techniques. In 2024 International Conference on Computing and Data Science (ICCDS) (pp. 1-6). IEEE.
[102] Cheng, N., Xie, H., Zhu, X., & Wang, H. (2023). Joint image enhancement learning for marine object detection in natural scene. Engineering Applications of Artificial Intelligence, 120, 105905.
[103] Zhu, P., Isaacs, J., Fu, B., & Ferrari, S. (2017, December). Deep learning feature extraction for target recognition and classification in underwater sonar images. In 2017 IEEE 56th annual conference on decision and control (CDC) (pp. 2724-2731). IEEE.
[104] Ge, H., Dai, Y., Zhu, Z., & Liu, R. (2022). A deep learning model applied to optical image target detection and recognition for the identification of underwater biostructures. Machines, 10(9), 809.
[105] Gašparović, B., Lerga, J., Mauša, G., & Ivašić-Kos, M. (2022). Deep learning

approach for objects detection in underwater pipeline images. Applied artificial intelligence, 36(1), 2146853.
[106] Jian, M., Yang, N., Tao, C., Zhi, H., & Luo, H. (2024). Underwater object detection and datasets: a survey. Intelligent Marine Technology and Systems, 2(1), 9.
[107] Wang, X., Han, T. X., & Yan, S. (2009, September). An HOG-LBP human detector with partial occlusion handling. In 2009 IEEE 12th international conference on computer vision (pp. 32-39). IEEE.
[108] Pang, Y., Yuan, Y., Li, X., & Pan, J. (2011). Efficient HOG human detection. Signal processing, 91(4), 773-781.
[109] Aguirre-Castro, O. A., Inzunza-González, E., García-Guerrero, E. E., Tlelo-Cuautle, E., López-Bonilla, O. R., Olguín-Tiznado, J. E., & Cárdenas-Valdez, J. R. (2019). Design and Construction of an ROV for Underwater Exploration. Sensors, 19(24), 5387.
[110] Fang, R., He, P., & Gao, Y. (2024, July). A review of SLAM techniques and applications in unmanned aerial vehicles. In Journal of Physics: Conference Series (Vol. 2798, No. 1, p. 012033). IOP Publishing.
[111] Wang, X., Fan, X., Shi, P., Ni, J., & Zhou, Z. (2023). An overview of key SLAM technologies for underwater scenes. Remote Sensing, 15(10), 2496.
[112] Rahman, S. (2020). A Multi-Sensor Fusion-Based Underwater Slam System (Doctoral dissertation, University of South Carolina).
[113] Hidalgo, F., & Bräunl, T. (2015, February). Review of underwater SLAM techniques. In 2015 6th International Conference on Automation, Robotics and Applications (ICARA) (pp. 306-311). IEEE.
[114] Zhang, S., Zhao, S., An, D., Liu, J., Wang, H., Feng, Y., ... & Zhao, R. (2022). Visual SLAM for underwater vehicles: A survey. Computer Science Review, 46, 100510.
[115] Ribas, D., Ridao, P., Tardós, J. D., & Neira, J. (2008). Underwater SLAM in man-made structured environments. Journal of Field Robotics, 25(11-12), 898-921.
[116] Guth, F., Silveira, L., Botelho, S., Drews, P., & Ballester, P. (2014, August). Underwater SLAM: Challenges, state of the art, algorithms and a new biologically-inspired approach. In 5th IEEE RAS/EMBS international conference on biomedical robotics and biomechatronics (pp. 981-986). IEEE.
[117] Rahman, S., Quattrini Li, A., & Rekleitis, I. (2022). SVIn2: A multi-sensor fusion-based underwater SLAM system. The International Journal of Robotics Research, 41(11-12), 1022-1042.
[118] Rahman, S. (2020). A Multi-Sensor Fusion-Based Underwater Slam System (Doctoral dissertation, University of South Carolina).
[119] Kok, M.; Solin, A. Scalable Magnetic Field SLAM in 3D Using Gaussian Process Maps. In Proceedings of the 21st International Conference on Information Fusion, Cambridge, UK, 10–13 July 2018; pp. 1353–1360.
[120] Djuric, P. M., Kotecha, J. H., Zhang, J., Huang, Y., Ghirmai, T., Bugallo, M. F., & Miguez, J. (2003). Particle filtering. IEEE signal processing magazine, 20(5), 19-38.
[121] Fareh, R., Khadraoui, S., Abdallah, M. Y., Baziyad, M., & Bettayeb, M. (2021). Active disturbance rejection control for robotic systems: A review. Mechatronics, 80, 102671.
[122] Boittiaux, C., Dune, C., Ferrera, M., Arnaubec, A., Marxer, R., Matabos, M., ... & Hugel, V. (2023). Eiffel Tower: A deep-sea underwater dataset for long-term visual localization. The International Journal of Robotics Research, 42(9), 689-699.

[123] Wang, R., Wang, X., Zhu, M., & Lin, Y. (2019). Application of a real-time visualization method of AUVs in underwater visual localization. Applied Sciences, 9(7), 1428.
[124] Carreras, M., Ridao, P., García, R., & Nicosevici, T. (2003, September). Vision-based localization of an underwater robot in a structured environment. In 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422) (Vol. 1, pp. 971-976). IEEE.
[125] Corke, P., Detweiler, C., Dunbabin, M., Hamilton, M., Rus, D., & Vasilescu, I. (2007, April). Experiments with underwater robot localization and tracking. In Proceedings 2007 IEEE International Conference on Robotics and Automation (pp. 4556-4561). IEEE.
[126] Trslić, P., Weir, A., Riordan, J., Omerdic, E., Toal, D., & Dooly, G. (2020). Vision-based localization system suited to resident underwater vehicles. Sensors, 20(2), 529.
[127] Nuske, S., Roberts, J., Prasser, D., & Wyeth, G. (2010, July). Experiments in visual localization around underwater structures. In Field and Service Robotics: Results of the 7th International Conference (pp. 295-304). Berlin, Heidelberg: Springer Berlin Heidelberg.
[128] dos Santos, M. M., de Oliveira Evald, P. J. D., De Giacomo, G. G., Drews-Jr, P. L. J., & da Costa Botelho, S. S. (2023). A probabilistic underwater localization based on cross-view and cross-domain acoustic and aerial images. Journal of Intelligent & Robotic Systems, 108(3), 34.
[129] Zhong, L., Li, D., Lin, M., Lin, R., & Yang, C. (2019). A fast binocular localization method for AUV docking. Sensors, 19(7), 1735.
[130] Qin, J., Li, M., Li, D., Zhong, J., & Yang, K. (2022). A survey on visual navigation and positioning for autonomous UUVs. Remote Sensing, 14(15), 3794.
[131] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306.
[132] Jung, J., Li, J. H., Choi, H. T., & Myung, H. (2017). Localization of AUVs using visual information of underwater structures and artificial landmarks. Intelligent Service Robotics, 10, 67-76.
[133] Zhao, C., Dong, H., Wang, J., Qiao, T., Yu, J., & Ren, J. (2023). Dual-Type Marker Fusion-Based Underwater Visual Localization for Autonomous Docking. IEEE Transactions on Instrumentation and Measurement.
[134] Zhang, P., Milios, E. E., & Gu, J. (2004, August). Underwater robot localization using artificial visual landmarks. In 2004 IEEE International Conference on Robotics and Biomimetics (pp. 705-710). IEEE.
[135] Buchan, A. D., Solowjow, E., Duecker, D. A., & Kreuzer, E. (2017, September). Low-cost monocular localization with active markers for micro autonomous underwater vehicles. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4181-4188). IEEE.
[136] Chavez, A. G., Mueller, C. A., Doernbach, T., & Birk, A. (2019). Underwater navigation using visual markers in the context of intervention missions. International journal of advanced robotic systems, 16(2), 1729881419838967.
[137] Wei, Q., Yang, Y., Zhou, X., Fan, C., Zheng, Q., & Hu, Z. (2023). Localization method for underwater robot swarms based on enhanced visual markers. Electronics, 12(23), 4882.
[138] Folkesson, J., Leederkerken, J., Williams, R., Patrikalakis, A., & Leonard, J.

(2008). A feature based navigation system for an autonomous underwater robot. In Field and Service Robotics: Results of the 6th International Conference (pp. 105-114). Springer Berlin Heidelberg.
[139] Murthy, C. B., Hashmi, M. F., Bokde, N. D., & Geem, Z. W. (2020). Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—A comprehensive review. Applied sciences, 10(9), 3280.
[140] Pyimage Research. Faster R-CNNs(2023). From: https://pyimagesearch.com/2023/11/13/faster-r-cnns/
[141] Github. Detectron 2. From: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md
[142] Wang, Y., Song, W., Fortino, G., Qi, L. Z., Zhang, W., & Liotta, A. (2019). An experimental-based review of image enhancement and image restoration methods for underwater imaging. IEEE access, 7, 140233-140251.
[143] Detectron 2. Detectron2 Model Zoo and Baselines. From: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md
[144] Medium. The learning rate: : A Hyperparameter That Matters. From: https://mohitmishra786687.medium.com/the-learning-rate-a-hyperparameter-that-matters-b2f3b68324ab
[145] CONRAD. Intel RealSense Depth Camera D435 Full HD Webcam 1920 x 1080 Pixel. From: https://www.conrad.nl/nl/p/intel-realsense-depth-camera-d435-full-hd-webcam-1920-x-1080-pixel-1707630.html?utm_source=google&utm_medium=surfaces&utm_campaign=shopping-feed&utm_content=free-google-shopping-clicks&utm_term=1707630&utm_source=google&utm_medium=cpc&utm_campaign=NL+-+PMAX+-+Nonbrand+-+HighSeller&utm_id=21918742254&gad_source=1&gclid=Cj0KCQjwkZm_BhDrARIsAAEbX1HQnoCVd-RTRNF9rnl2qZBMwFvfKrjKa-pE4r6FJ8NG5duPlvWCa6EaAgr-EALw_wcB
[146] TREATSTOCK. TPU. From: https://zh.treatstock.com/material/tpu
[147] SOLAXIS. PLA. From: https://solaxis.ca/en/material/pla-industrial-3d-printing/
[148] Islam, M. J., Xia, Y., & Sattar, J. (2020). Fast underwater image enhancement for improved visual perception. IEEE Robotics and Automation Letters, 5(2), 3227-3234.
[149] Minnesota. The EUVP dataset. From: https://irvlab.cs.umn.edu/resources/euvp-dataset
[150] Er, Meng Joo, et al. "Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review." Sensors 23.4 (2023): 1990.
[151] Jian, M., Yang, N., Tao, C., Zhi, H., & Luo, H. (2024). Underwater object detection and datasets: a survey. Intelligent Marine Technology and Systems, 2(1), 9.
[152] Roboflow. From: https://universe.roboflow.com/
[153] Framos. The working principle of Depth-Sensing Cameras. From: https://www.framos.com/en/articles/what-are-depth-sensing-cameras-and-how-do-they-work
[154] Intel REALSENSE. Depth Camera D455. From: https://www.intelrealsense.com/depth-camera-d455/
[155] Intel REALSENSE. Depth Camera D456. From: https://www.intelrealsense.com/depth-camera-d456/
[156] Intel REALSENSE. Start building your own depth applications. From:

https://www.intelrealsense.com/developers/
[157] Luxonis. OAK-D. From: https://shop.luxonis.com/products/oak-d?srsltid=AfmBOopG-8acmmSZbHhQz0IJTnNeTl5vayJ9Txt-xBoqIqEUOrZv-FUq
[158] Github. depthai-experiments. From: https://github.com/luxonis/depthai-experiments
[159] Ultralytics. YOLOv5. From: https://docs.ultralytics.com/models/yolov5/
[160] Metric3d v2: A versatile monocular geometric foundation model for zero-shot
[161] Github. MiDas. From: https://github.com/isl-org/MiDaS
[162] Google Colab. MiDaS. From: https://colab.research.google.com/drive/1dJpTQmHHFj0qxYLMaUK_dPNNnPiNVYDn#scrollTo=c76d3796

Acknowledgement

I would like to express my deepest and most sincere gratitude to my supervisors, Dr. Jovana Jovanova, Dr. Pooria Pahlavan, and Filippo Riccioli, for their invaluable guidance, constant support, and insightful feedback throughout the course of this project. Their expertise, encouragement, and patience have been instrumental in shaping both the direction and quality of this research. I have learned a great deal under their mentorship, and I am truly grateful for the opportunity to work with such dedicated and inspiring researchers.

My heartfelt appreciation also goes to the Faculty Workshop of Mechanical Engineering and Ton Veer for their generous technical assistance and collaborative spirit. Your practical insights and hands-on support significantly broadened my understanding of the subject, and without your expertise, the fabrication of components and implementation of the experimental setup would have been far more challenging.

I would also like to extend my thanks to my colleagues Ruochen Wu, Aaron Chen and all members in the research group, whose support was vital in carrying out this research. Your willingness to share tools and facilities, as well as your thoughtful advice and creative problem-solving ideas, contributed meaningfully to overcoming several technical challenges. I truly appreciated the team spirit and open exchange of knowledge that made our collaboration so effective.

Special thanks go to my family and friends, who have stood by me unwaveringly through the highs and lows of this journey. Your emotional encouragement, moral support, and, most importantly, financial assistance have made it possible for me to pursue and complete my studies. Your belief in me, even during the most difficult moments, was a constant source of strength and motivation.

Finally, I am deeply grateful to everyone—mentioned and unmentioned—who has contributed to this work in one way or another. Whether through academic guidance, technical help, or personal support, your contributions have been crucial.

Enrong Xiang
Delft, May 2025