Vision-Based Underwater Docking Station Detection and Pose Estimation for Autonomous Underwater Vehicle (AUVs)

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Bachelor of Technology

by

BISHAL HAZARIKA (200101034)
DHRUBA JYOTI SARMA (200101003)

Under the guidance of

Dr. Anjan Kumar Talukdar
Prof. Kandarpa Kumar Sarma
and
Under the co-guidance of
Dr. Surajit Deka

Department of Electronics and Communication Engineering
GAUHATI UNIVERSITY
July, 2024

Salutation to the noble Guru, who is Brahma, Vishnu and Maheswara,
the direct Parabrahma, the Supreme Reality.

D E C L A R A T I O N

I hereby declare that the work described in the thesis entitled **"Vision-Based Underwater Docking Station Detection and Pose Estimation for Autonomous Underwater Vehicle (AUVs)"** is an original work done under the guidance of Dr. Anjan Kumar Talukdar, Assistant Professor, Department of Electronics and Communication Engineering, Gauhati University, Guwahati-14 and this work has not been submitted elsewhere for award of any degree.

Bishal Hazarika (200101034) and Dhruba Jyoti Sarma (200101003)
B.Tech, 8th Semester
Department of Electronics and Communication Engineering,
Gauhati University
Guwahati-781014, Assam, India
July, 2024

C E R T I F I C A T E

The work contained in the project titled **"'Vision-Based Underwater Docking Station Detection and Pose Estimation for Autonomous Underwater Vehicle (AUVs)"** by **Bishal Hazarika** (200101034) and **Dhruba Jyoti Sarma** (200101003) , has been carried out under my supervision in Department of Electronics and Communication Engineering as part of B.Tech in Electronics and Communication Engineering, programme and the contents derived out of original work carried out during this period at the Department of Electronics and Communication Engineering and has not been submitted elsewhere for a degree to the best of my knowledge.

.....

Prof. Kandarpa Kumar Sarma
Professor and Head
Department of Electronics and
Communication Engineering,
Gauhati University
Guwahati-781014, Assam, India
3rd July, 2024

.....

Dr. Anjan Kumar Talukdar
Assistant Professor
Department of Electronics and
Communication Engineering,
Gauhati University
Guwahati-781014, Assam, India
3rd July, 2024

Project Evaluation Sheet

Certified that , completed and presented the work done titled **"Vision-Based Underwater Docking Station Detection and Pose Estimation for Autonomous Underwater Vehicle (AUVs)"** and submitted a thesis with identical name and was evaluated to fulfill partial requirements of the Bachelor of Technology (B.Tech) degree in Electronics and Communication Engineering programme of Gauhati University, Guwahati-781014, Assam, India.

Supervisors:

1. Name: Dr. Anjan Kumar Talukdar

Signature:.....

2. Name: Prof. Kandarpa Kr. Sarma

Signature:.....

Co-Supervisors:

1. Name: Dr. Surajit Deka

Signature:.....

Internal Examiners:

Signature:.....

External Examiner:

Name:.....

Department:.....

Signature:.....

Head of the Department:

Name: Prof. Kandarpa Kr. Sarma

Signature:.....

Date: 3rd July, 2024

Acknowledgments

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people made it possible, whose constant guidance and encouragement crown all the efforts. An understanding of the study like this, is never the outcome of the effort of a single person, rather it bears the imprint of a number of people who directly or indirectly helped us in partial fulfillment of the project. We would be failing in our duty if we do not say a word of thanks to all these people.

First and foremost we would like to gratefully acknowledge our gratitude to Dr. Anjan Kumar Talukdar, our project guide and Assistant Professor of Electronics and Communication Engineering, Gauhati University, for his inspiring advise, supervision and guidance. His truly scientific intuition has made him as a constant oasis of ideas and passions in science, which exceptionally inspired and enriches our growth as a student and researcher. As a guide his enthusiastic attitude inspired us a lot. The work culture that has been nurtured by him in the lab is worth noticing, which help us in building up our working attitude. The freedom that he has provided us in carrying out our research work helped in developing the confidence for an independent researcher.

Date: _____

BISHAL HAZARIKA (200101034)
DHRUBA JYOTI SARMA (200101003)

Abstract

This work presents a comprehensive study on Vision-Based Underwater Docking Station Detection and Pose Estimation for Autonomous Underwater Vehicles (AUVs). The primary challenge addressed in this research is the robust detection of docking stations in underwater environments, which are characterized by conditions such as turbid water, light scattering, low visibility in deep oceanic regions, and environmental pollution. To overcome these challenges, the project focuses on developing a detection algorithm capable of excelling in adverse underwater conditions. Furthermore, recognizing the difficulty in collecting ample underwater data for training purposes, the research explores methods to achieve high detection accuracy with limited data.The project leverages state-of-the-art detection algorithms, including YOLO v8 and Detection Transformer. The latter, a cutting-edge technology, is explored for its potential to provide superior results compared to traditional methods like YOLO. Through rigorous experimentation, the study aims to validate the efficacy of these algorithms in enhancing the accuracy of underwater docking station detection.In the second phase of the project, the focus shifts to the pose estimation of the detected docking stations by the AUV using computer vision techniques. This involves the application of camera calibration and pose estimation methodologies to facilitate precise docking maneuvers. The ultimate objective is to enable AUVs to autonomously navigate and dock accurately with underwater stations, contributing to advancements in autonomous underwater exploration and operations.The outcomes of this research hold promise for applications in various fields, including marine exploration, environmental monitoring, and underwater infrastructure maintenance, where reliable docking station detection and pose estimation are critical for the success of autonomous underwater missions.

List of Abbreviation

- AUV - Autonomous Underwater Vehicle
- CNN - Convolutional Neural Network
- YOLO - You Only Look Once
- DETR - Detection Transdormer
- RT DETR - Real Time Detection Transdormer
- mAP50 - mean Average Precision with IoU threshold of 0.5
- mAP50-95 - mean Average Precision with IoU threshold ranging from 0.5 to 0.95
- IoU - Intersection Over Union
- val - Validation
- AP - Average Precision
- CUDA - Compute Unified Device Architecture
- FPS - Frames Per Second
- GPU - Graphics Processing Unit
- HTC - Hough Circle Transform

Contents

Abstract v

Abbreviation vi

List of Tables ix

List of Figures x

1 Introduction 1

1.1 Background 1

1.2 Motivation 2

1.3 Literature Review 2

1.4 Problem Formulation 6

1.5 Contribution 6

1.6 Organization of the Thesis 6

2 Theoretical Background 8

2.1 Introduction 8

2.2 Technical Aspects of Computer Vision 8

2.3 Convolutional Neural Networks (CNN) 9

2.4 Introduction to Object Detection 12

2.4.1 Evolution of Object Detection Algorithms 13

2.4.2 You Only Look Once (YOLO) 14

2.4.3 YOLO v8 16

2.4.4 Real Time - Detection Transformer (RT-DETR) 16

2.5 Pose Estimation 18

2.6 Conclusion 19

3 Underwater Docking Station Detection Using Deep Learning Models 20

3.1 Introduction 20

3.2 Methodology 21

3.2.1 Data Acquisition 21

3.2.2 Data Annotation and Labelling 22

3.2.3 Pre-Processing Dataset 23

3.2.4 Feeding Data to the YOLO v8 Model 24

3.2.5 Feeding Data to RT-DETR Model 25

3.2.6 Block Diagram 26

3.3 Results 26

3.3.1 Deep Learning Models Outputs 26

3.4 Conclusion 29

4 Pose Estimation of Underwater Docking Station by the Automated Underwater Vehicles (AUVs) using Vision based Algorithm 30

4.1 Introduction 30

4.2 Methodology 31

4.2.1 Camera Calibration 31

4.2.2 Types of Calibration methods 32

4.2.3 Checkerboard Pattern 33

4.2.4 Pose Estimation 35

4.2.5 Contours Thresholding and Circle Fitting 36

4.2.6 3 Dimensional Distance Calculation 37

4.2.7 Docking Logic 38

4.3 Results 39

4.4 Conclusion 41

5 Conclusion and Future Direction 43

5.1 Summary of the work done 43

5.2 Limitation 44

5.3 Future Direction 45

List of Tables

1.1 Literature Review 3

3.1 Mertrics Comparison between YOLO v8 and RT-DETR 28

4.1 Comparison between actual distance and calculated distance 41

List of Figures

1.1 Docking of an AUV 1

2.1 Architecture of CNN [9] 10
2.2 Convolution Operation 11
2.3 Pooling Operation 11
2.4 Architecture of YOLO [11] 14
2.5 Quantitative Comparison of YOLO with other algorithms on three datasets [11] 15
2.6 Architecture of DETR [12] 16
2.7 Compared to other real-time object detectors, RT DETR achieves state-of-the-art performance in both speed and accuracy [13] 17
2.8 Architecture of RT-DETR [13] 17

3.1 Image Labelling (*source* : Roboflow) 22
3.2 Overview of the Dataset (*source* : Roboflow) 23
3.3 Image Augmentation 24
3.4 Block Diagram : Docking station detection 26
3.5 Epoch vs Mean Average Precision : Yolo v8 27
3.6 Epoch vs Mean Average Precision : RT- DETR 27
3.7 Confusion Matrix (*left*: YOLOv8 , *right* : RT-DETR) 28

4.1 Captured image of Checkerboard Pattern 34
4.2 Result after drawing the detected checker board corners 34
4.3 Visual Representation of Pose Estimation of AUV 35
4.4 Contour Detection 36
4.5 Circle Fitting 37
4.6 Guidance with arrow for center alignment 39

4.7 Centre perfectly aligned (X and Y distances = 0 cm) 40
4.8 AUV ready to dock at 25 cm after centre alignment 40

Chapter 1

Introduction

1.1 Background

Underwater docking is a critical task for many underwater operations, such as exploration, maintenance, and construction. However, it can be challenging and dangerous, especially in turbid water and low visibility conditions. Traditional underwater docking methods rely on human intervention and specialized equipment, which can be expensive and time-consuming. Vision-based underwater docking station detection and pose estimation systems offer a promising alternative to traditional methods. The system use cameras to capture images of the docking station and then use computer vision algorithms to detect and estimate the pose of the docking station relative to the AUV. This information can then be used to guide the AUV to the docking station in a safe and efficient manner. An AUV heading towards the docking station is shown in figure 1.1

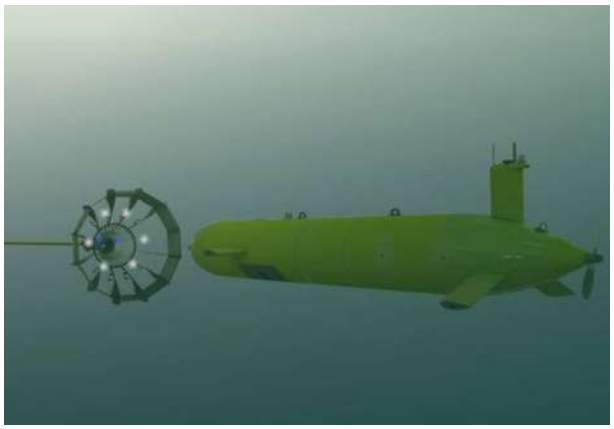

Figure 1.1: Docking of an AUV

1.2 Motivation

Our project addresses the urgent need for efficient underwater docking solutions, driven by the challenges of deep ocean operations.

Enhanced Efficiency and Safety: Traditional underwater docking methods depend on human intervention and specialized equipment, which are costly and time-consuming. Vision-based systems automate the docking process, significantly improving efficiency and safety.

Expanded Operational Range: These systems enable Autonomous Underwater Vehicles (AUVs) to function in challenging environments, such as turbid waters with low visibility, thereby broadening their application scope.

Reduced Human Intervention: By minimizing the need for human involvement in docking tasks, vision-based systems allow operators to focus on monitoring AUV operations or managing multiple AUVs simultaneously.

Support for New Applications: Vision-based docking technology fosters the development of new underwater applications, including exploration, maintenance, and construction.

1.3 Literature Review

There are many research were done on vision based object detection and pose estimation. Many optimization techniques are used in recent time to optimise it. Some of the research papers which we used as reference are discussed below:

Paper 1: This study introduces a method for image detection using YOLO in OpenCV, highlighting substantial improvements in both detection speed and accuracy. The implementation process is described in detail, showcasing the benefits of YOLO for real-time object detection in various scenarios.[1]

Paper 2: The authors evaluate the limitations of current top-tier object detection algorithms in underwater environments. They provide an in-depth analysis and suggest enhancements to improve detection accuracy and reliability under these challenging conditions.[2]

Paper 3: This research investigates the use of transfer learning with YOLO on ROV camera data for underwater object detection, achieving a 4% increase in mAP. The study demonstrates the effectiveness of transfer learning techniques in improving the vision systems of underwater robots.[3]

Paper 4: This paper reviews recent deep learning methods for underwater image analysis, discussing their principles, strengths, weaknesses, and applications. It also examines the technical challenges in AUV detection of underwater threats and suggests solutions to enhance detection accuracy and reliability.[4]

Paper 5: For reliable detection of docking stations, the authors propose a CNN named docking neural network (DoNN). They integrate a perspective-n-point algorithm for precise pose estimation, thereby improving the accuracy of underwater docking operations.[5]

Paper 6: Experimental results show an absolute estimation error of less than 10% for each degree of freedom in the PPARD. The new extraction method achieves an 87.99% success rate, outperforming classical methods and enhancing AUV positioning accuracy relative to docking stations.[6]

Paper 7: The authors present a system using cameras and active-light markers to determine relative positions. They provide a complete mathematical model for the positioning system, ensuring accurate and reliable pose estimation for extended under-ice AUV operations.[7]

Paper 8: This paper presents a computer vision-based method using deep learning to improve AUV positioning. The method reduces positioning errors to 30–60 meters, making it suitable for low-cost vehicles and enhancing navigation accuracy in marine environments.[8]

Table 1.1: Literature Review

.

SlNo	**Title**	**Author**	**Contributions**
1	YOLO Object Detection Using OPENCV	Akshara Gupta, Aditya Verma, Aditya Yadav, Arvindhan M	This paper gives a efficient approach for image detection using YOLO in OpenCV[1]

.

2	Underwater Object Classification and Detection	Andre Jesus, Claudio Zito, Claudio Tortorici, Eloy Roura, Giulia De Masi	This paper's key contribution is to analyze and quantify the shortcomings of conventional state-of-the-art algorithms for object detection in underwater environments.[2]
3	Object Detection using Transfer Learning for Underwater Robot	Chia-Chin Wang, Hooman Samani	This study explores transfer learning for underwater object detection using YOLO on ROV camera data. It achieves a 4% mAP improvement, demonstrating feasibility for underwater robots.[3]
4	Underwater target recognition methods based on the framework of deep learning	Bowen Teng and Hongjian Zhao	The paper surveys recent deep learning techniques for underwater image analysis, explaining principles, strengths/weaknesses, and applicability. It then analyzes technical challenges in AUV detection of underwater threats and proposes solutions.[4]
5	Detection and Pose Estimation for Short-Range Vision-Based Underwater Docking	Shuang Liu , Mete Ozay , Takayuki Okatani, Hongli Xu, Kai Sun, Yang Lin	For robust and credible detection of docking stations, a CNN called docking neural network (DoNN)is used. For accurate pose estimation, a perspective-n-point algorithm is integrated into the framework[5]

6	Estimation of Positions and Poses of Autonomous Un- derwater Vehicle Relative to Docking Station Based on Adaptive Extraction of Visual Guidance Features	Fengtian Lv, Huixi Xu, Kai shi, Xiaohui Wang	Experimental results demonstrated less than 10% absolute estimation error for each degree of freedom in the PPARD.The new extraction method achieved an 87.99% success rate, surpassing classical methods.[6]
7	Pose estimation for the case of prolonged under ice based AUV	D Frolov , S Polovko	Provides a system for determining the relative position based on cameras and active-light markers. A system equation and a complete mathematical modeling for the positioning system is performed. [7]
8	Computer Vision-Based Position Estimation for an Autonomous Underwater Vehicle	Jacek Zalewski, Stanisław Hożyń	AUVs, rapidly advancing in marine technology, face navigation issues due to electronic equipment weaknesses. This paper introduces a computer vision-based method using deep learning for improved AUV positioning, reducing errors to 30–60 meters, suitable for low-cost vehicles. [8]

1.4 Problem Formulation

The following problems are identified after having a detailed survey of the papers.

- Designing an Efficient Underwater Docking Station Detection model with limited dataset using Transformer based object detection.
- Pose Estimation of the Docking Station for proper and accurate docking of the AUV.

1.5 Contribution

The main objectives of this work are mentioned below:

1. Built a system that can find and position an underwater docking station using cameras.
2. The system is more accurate than other systems that have been used for this task.
3. The system can work in difficult conditions, such as turbid water and low visibility.
4. The system could help robots work better underwater, which could be useful for things like exploring the ocean or fixing things underwater.[2]

1.6 Organization of the Thesis

Chapter 1 provides comprehensive insights into the background, motivation, and problem formulation of the research work. It delves into the key factors that have influenced the project, highlighting the need for an efficient docking station detection algorithm and a accurate pose estimation of the station by the AUVs. We have conducted an extensive review of relevant literature, which is presented in a well-organized tabular format.

Chapter 2 We go into the theoretical underpinnings of computer vision, delving into the complexities of object detection and analysing sophisticated algorithms like Detection Transformer and Yolo v8. This chapter also explores the methods used for Pose Estimation.

Chapter 3 focuses on the development of an innovative object detection algorithm designed for the identification of docking stations for Autonomous Underwater Vehicles (AUVs). The

chapter outlines the utilization of Real-Time Detection Transformer (RT-DETR) and includes a comprehensive comparison with state-of-the-art algorithms like Yolo v8. The primary goal is to enhance the efficiency of AUVs docking procedures through the real-time detection capabilities of RT-DETR.

Chapter 4 Pose estimation for AUV docking stations is explored, with a focus on the need of accurate positioning for effective AUV docking. To ensure that the AUVs are properly aligned with the docking station, precise posture estimation is achieved through the use of vision-based algorithms. The approaches, conclusions, and outcomes of the pose estimation process are described in depth in this chapter, which enhances the overall efficacy of AUV docking systems.

Chapter 5 is the summery, limitations of our project and discusses the possible future research.

Chapter 2

Theoretical Background

2.1 Introduction

Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world. Using digital images from cameras and videos and deep learning models, machines can accurately identify and classify objects — and then react to what they "see." The goal of computer vision is to mimic the abilities of human vision by electronically perceiving and understanding an image. It is a multidisciplinary field that could broadly be called a subfield of AI and machine learning, but it intersects with data science, robotics, IoT, and more.

Computer vision tasks include methods for acquiring, processing, analyzing, and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information.

2.2 Technical Aspects of Computer Vision

Computer vision is a complex field that involves a series of steps to convert visual data into understandable information. Here are some of the technical aspects involved in computer vision:

Image Processing: This is the first step in the computer vision pipeline. It involves enhancing the image quality by removing noise, adjusting brightness/contrast, and other such operations. Techniques used include filtering, edge detection, and color space conversion.

Feature Extraction: This involves identifying key points or features in an image that are

unique and can be used for further analysis. Techniques used include SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features), and HOG (Histogram of Oriented Gradients).

Object Detection: This involves identifying and locating objects of interest in an image. Techniques used include R-CNN (Region-based Convolutional Neural Networks), Fast R-CNN, and Faster R-CNN.

Image Segmentation: This involves partitioning an image into multiple segments or "superpixels". Techniques used include semantic segmentation, instance segmentation, and watershed transformation.

Object Recognition: This involves identifying the class of an object in an image. Techniques used include CNN (Convolutional Neural Networks), k-NN (k-Nearest Neighbors), and SVM (Support Vector Machines).

Scene Reconstruction: This involves creating a 3D model of a scene from a series of 2D images. Techniques used include stereo vision, structure from motion (SfM), and volumetric reconstruction.

Image Registration: This involves aligning two or more images of the same scene taken at different times, from different viewpoints, or by different sensors. Techniques used include feature-based registration, intensity-based registration, and edge-based registration.

2.3 Convolutional Neural Networks (CNN)

A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be. The human brain processes a huge amount of information the second we see an image. Each neuron works in its own receptive field and is connected to other neurons in a way that they cover the entire visual field. Just as each neuron responds to stimuli only in the restricted region of the visual field called the receptive field in the biological vision system, each neuron in a CNN processes data only in its receptive field as well. The layers are arranged in such a way so that they detect simpler patterns first (lines, curves, etc.) and more complex patterns (faces, objects, etc.) further along. By using a CNN, one can enable sight to computers.

CNN have proven highly effective in tasks such as image classification, object detection, and facial recognition due to their ability to capture hierarchical patterns and spatial relationships within images.

The Architechture of CNN:

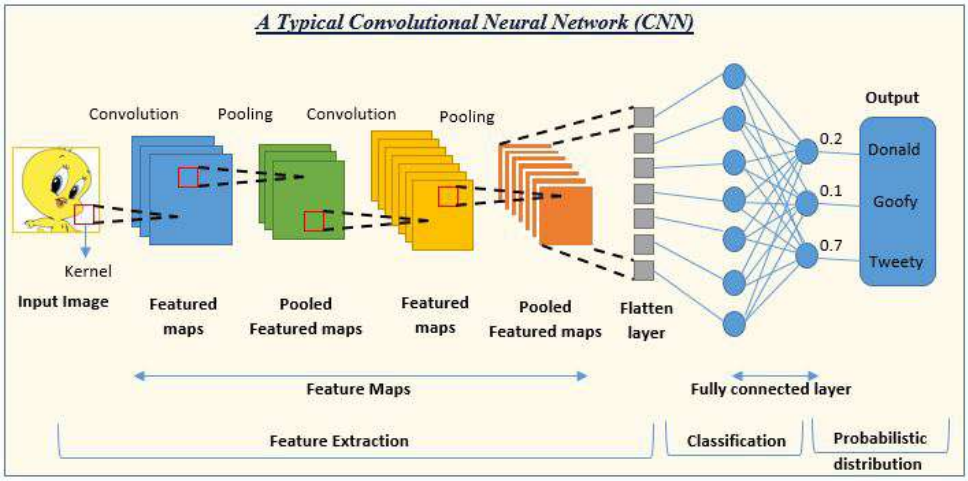

Figure 2.1: Architecture of CNN [9]

The Architecture of CNN is shown in Figure 2.1

1. **Input Layer:** The input layer of a CNN receives the raw pixel values of the image being processed. Each pixel serves as an individual input node, forming the initial data that the network processes.

2. **Convolutional Layer:** The Convolutional Layer is the first layer in a CNN. It performs a convolution operation by applying a set of filters or kernels to the input image.The convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load.

This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in-depth. This means that, if the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all three channels.

These filters are designed to detect specific features such as edges, textures, or more complex patterns. The output of this layer is a feature map, a representation of the spatial presence of

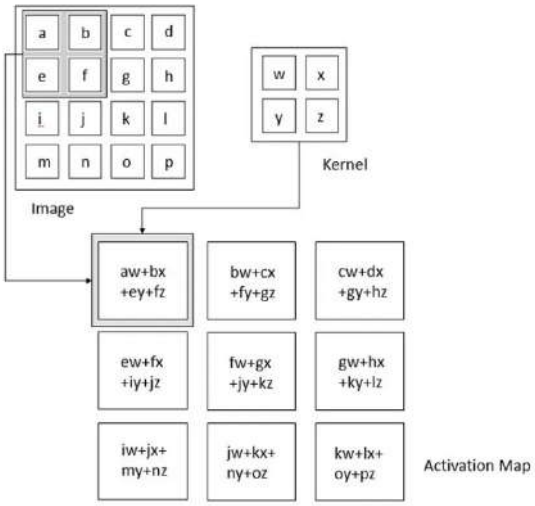

Figure 2.2: Convolution Operation

features detected by the filters. Convolution allows the network to identify local patterns in different parts of the image. The Convolution Operation is shown in Figure 2.2

3. **Pooling Layer:** The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This helps in reducing the spatial size of the representation, which decreases the required amount of computation and weights. The pooling operation is processed on every slice of the representation individually.

There are several pooling functions such as the average of the rectangular neighborhood, L2 norm of the rectangular neighborhood, and a weighted average based on the distance from the central pixel. However, the most popular process is max pooling, which reports the maximum output from the neighborhood. Figure 2.3 shows the Pooling Operation.

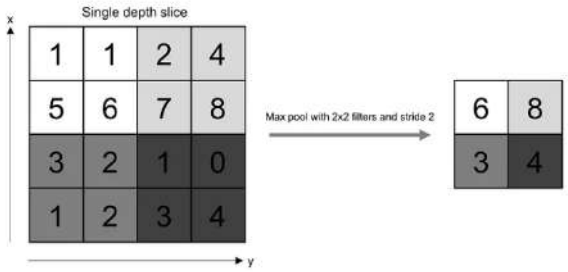

Figure 2.3: Pooling Operation

4. **Flatten Layer:** After the Convolutional, ReLU, and Pooling layers, a Flatten Layer is often used. This layer transforms the multi-dimensional output into a one-dimensional vector,

preparing the data for input into the Fully Connected Layer.

5. **Fully Connected Layer:** The Fully Connected Layer takes the flattened output from the previous layers and uses it for classification. Neurons in this layer are connected to every neuron in the preceding layer, allowing the network to learn complex relationships and make predictions based on the learned features. In the context of image classification, the Fully Connected Layer outputs probabilities for each class, and the class with the highest probability is considered the final prediction. [9]

2.4 Introduction to Object Detection

Object detection is a computer vision technique for locating instances of objects in images or videos. It's a key technology behind advanced driver assistance systems (ADAS) that enable cars to detect driving lanes or perform pedestrian detection to avoid accidents. Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results. When humans look at an image, we can recognize and locate objects of interest rapidly. Object detection algorithms aim to achieve the same task, but automatically and more accurately.

There are several ways to perform object detection using computer vision, all of which involve some form of machine learning. At a high level, these are the main steps involved in the object detection process:

Training: First, a model is trained using a large dataset of images that are labeled with bounding boxes and class names. The model learns to recognize the appearance of the classes it has been trained on.

Sliding Window Detection: The trained model is run on a new image and makes predictions at different locations and scales. High scoring regions of the image are considered detections.

Non-Maximum Suppression: The final step in the detection pipeline is to resolve overlapping detections. This is typically achieved by "suppressing" or removing detections that have a high overlap with a detection of higher score.

In the next section, we will delve deeper into the evolution of object detection algorithms, from R-CNN to the latest YOLO v8 and RTDETR.

2.4.1 Evolution of Object Detection Algorithms

Object detection has been widely used for face detection, vehicle detection, pedestrian counting, web images, security systems and driverless cars. Let's look at the evolution of object detection algorithms:

1. **R-CNN (Regions with CNN features):** R-CNN creates bounding boxes, or region proposals, using a process called Selective Search and then run a convolutional neural network (CNN) on these proposed boxes. After running the CNN, it uses SVM (Support Vector Machine) to classify the objects and a linear regression model to tighten the bounding boxes. However, it was slow due to running a CNN for every individual region proposal.

2. **Fast R-CNN:** Fast R-CNN introduced RoI (Region of Interest) pooling, which allowed the network to reuse the computations from the forward pass. It was faster than R-CNN and also solved another problem of updating the weights of the CNN and SVM separately.

3. **Faster R-CNN:** Faster R-CNN introduced RPN (Region Proposal Network) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. It is a fully end-to-end method for object detection.

4. **YOLO (You Only Look Once):** YOLO framed object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. It looks at the whole image only once and is super fast, but less accurate especially for small objects.

5. **YOLO v2, v3, v4:** These versions improved upon the original YOLO by introducing anchor boxes to deal with different shapes, and various architecture changes to improve speed and accuracy.[10]

6. **YOLO v8:** The latest version as of now, it introduced transformer encoders in place of the traditional convolution layers, making it even more accurate.

7. **RTDETR (Real-Time DETR):** This is a real-time version of DETR (DEtection TRansformer). It uses a transformer encoder-decoder architecture, eliminating the need for anchor boxes and providing a potential new direction for object detection algorithms.

2.4.2 You Only Look Once (YOLO)

YOLO (You Only Look Once) is a real-time object detection system that is both fast and accurate. It is a unified model that can detect multiple objects in an image, and it is trained on a loss function that directly corresponds to detection performance. Unlike other object detection methods, YOLO can be trained directly on full images, making it simpler to construct and easier to use.

YOLO Architecture

The YOLO model is made up of three key components: the head, neck, and backbone. The backbone is the part of the network made up of convolutional layers to detect key features of an image and process them. The backbone is first trained on a classification dataset, such as ImageNet, and typically trained at a lower resolution than the final detection model, as detection requires finer details than classification. The neck uses the features from the convolution layers in the backbone with fully connected layers to make predictions on probabilities and bounding box coordinates. The head is the final output layer of the network which can be interchanged with other layers with the same input shape for transfer learning. As discussed earlier, the head is an S × S × (C + B 5) tensor and is 7 × 7 × 30 in the original YOLO research paper with a split size S of 7, 20 classes C, and 2 predicted bounding boxes B. These three portions of the model work together to first extract key visual features from the image then classify and bound them. The Architecture of YOLO is shown in Figure 2.4

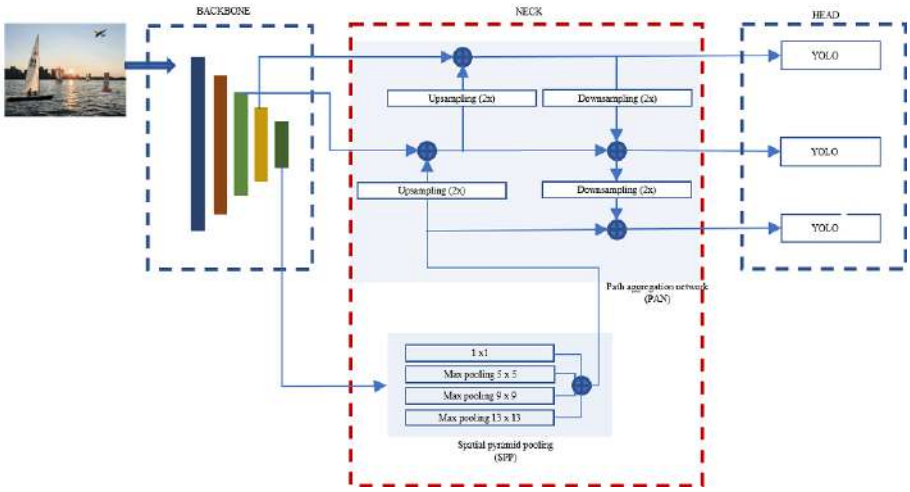

Figure 2.4: Architecture of YOLO [11]

Performance of YOLO

In terms of speed, YOLO is one of the fastest general-purpose object detectors in the literature. It can process images in real-time, with a frame rate of up to 45 frames per second on a Titan X GPU. This makes it ideal for applications that require fast, robust object detection, such as self-driving cars, surveillance systems, and robotics.

In terms of accuracy, YOLO is also highly effective. It outperforms other real-time detectors such as DPM and Fast R-CNN, and it is competitive with state-of-the-art detectors such as Faster R-CNN and SSD. YOLO is particularly good at detecting small objects and objects in cluttered scenes, which are challenging for other detectors.

	VOC 2007	Picasso		People-Art
	AP	AP	Best F_1	AP
YOLO	**59.2**	**53.3**	**0.590**	**45**
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	

Figure 2.5: Quantitative Comparison of YOLO with other algorithms on three datasets [11]

A quantitative comparison of YOLO with other algorithms is shown in Figure 2.5

The YOLO object detection algorithm outperforms the other algorithms on the VOC 2007, Picasso and People-Art Datasets in terms of Mean Average Precision (AP) as shown in 2.5

One of the key advantages of YOLO is its ability to generalize well to new domains. It has been tested on a variety of datasets, including the PASCAL VOC dataset and two artwork datasets, and it has consistently outperformed other detectors. This makes it a versatile tool for a wide range of applications.

Overall, YOLO is a highly effective and efficient object detection system that is well-suited for real-time applications. Its speed and accuracy make it a valuable tool for a variety of industries, and its ability to generalize to new domains makes it a versatile and flexible solution.[11]

2.4.3 YOLO v8

YOLOv8 is the latest version of the acclaimed real-time object detection and image segmentation model. YOLOv8 is built on cutting-edge advancements in deep learning and computer vision, offering unparalleled performance in terms of speed and accuracy. Its streamlined design makes it suitable for various applications and easily adaptable to different hardware platforms, from edge devices to cloud APIs.

2.4.4 Real Time - Detection Transformer (RT-DETR)

DETR (DEtection TRansformer) is a relatively new object detection algorithm that was introduced in 2020 by researchers at Facebook AI Research (FAIR). It is based on the transformer architecture, a powerful sequence-to-sequence model that has been used for various natural language processing tasks. Traditional object detectors (i.e., R-CNN and YOLO) are complex and have gone through multiple variations and rely on hand-designed components (i.e., NMS). DETR, on the other hand, is a direct set prediction model that uses a transformer encoder-decoder architecture to predict all objects at once. This approach is simpler and more efficient than traditional object detectors and achieves comparable performance on the COCO dataset. Figure 2.6 shows the Architecture of DETR.

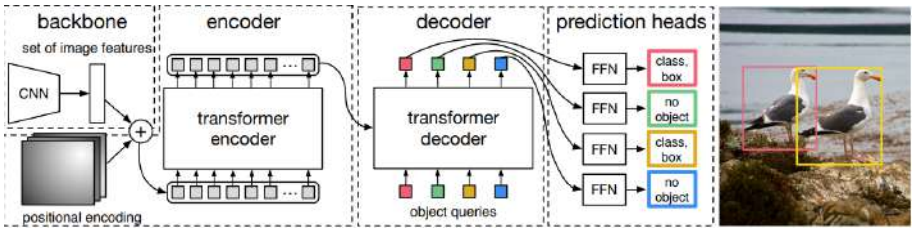

Figure 2.6: Architecture of DETR [12]

The DETR architecture is simple and consists of three main components: a CNN backbone (i.e., ResNet) for feature extraction, a transformer encoder-decoder, and a feed-forward network (FFN) for final detection predictions. The backbone processes the input image and generates an activation map. The transformer encoder reduces the channel dimension and applies multi-head self-attention and feed-forward networks. The transformer decoder uses parallel decoding of N object embeddings and independently predicts box coordinates and class labels using object queries. DETR reasons about all objects together using pair-wise relations, benefiting from the whole image context.[12]

Real-Time Detection Transformer (RT-DETR), is a cutting-edge end-to-end object detector that provides real-time performance while maintaining high accuracy. It leverages the power of Detection Transformers (DETR) to efficiently process multiscale features by decoupling intra-scale interaction and cross-scale fusion. RT-DETR is highly adaptable, supporting flexible adjustment of inference speed using different decoder layers without retraining. The model excels on accelerated backends like CUDA with TensorRT, outperforming many other real-time object detectors like YOLO. Figure 2.7 shows the comparison of various real-time object detectors. The Architecture of RT-DETR is shown in Figure 2.8. The proposed Real-Time

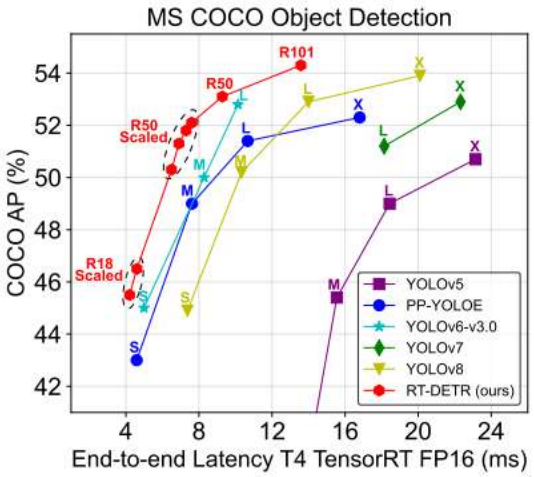

Figure 2.7: Compared to other real-time object detectors, RT DETR achieves state-of-the-art performance in both speed and accuracy [13]

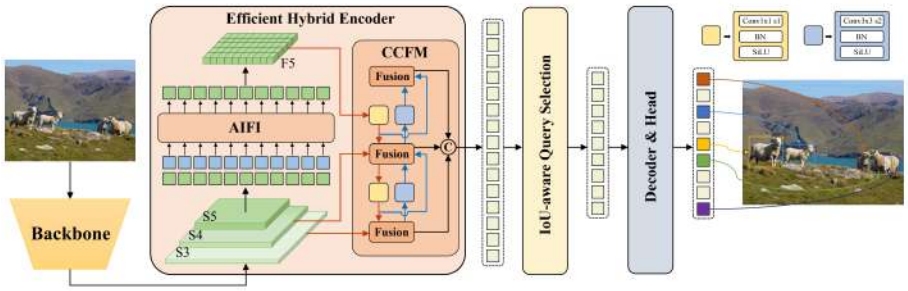

Figure 2.8: Architecture of RT-DETR [13]

DEtection TRansformer (RT-DETR) is a pioneering model designed to address the computational cost limitations of traditional DETRs while achieving real-time object detection. Here's a breakdown of how the model works:

1. **Backbone**: RT-DETR leverages a backbone, which consists of the output features of the last

three stages of the backbone S3, S4, S5, as the input to the encoder.

2. **Hybrid Encoder**: The hybrid encoder plays a crucial role in transforming multi-scale features into a sequence of image features through intra-scale interaction and cross-scale fusion. This process efficiently processes features with different scales, reducing unnecessary computational redundancy ,[object Object],.

3. **IoU-aware Query Selection**: The model employs IoU-aware query selection to choose a fixed number of image features from the encoder output sequence to serve as initial object queries for the decoder. This selection process is crucial for optimizing object queries to generate boxes and confidence scores ,[object Object],.

4. **Transformer Decoder with Auxiliary Prediction Heads**: The decoder, equipped with auxiliary prediction heads, iteratively optimizes object queries to generate boxes and confidence scores. This iterative optimization process is essential for accurate and efficient object detection.

By combining these components, RT-DETR achieves real-time object detection capabilities while effectively addressing the computational cost limitations of traditional DETRs. The model's innovative design and efficient processing of multi-scale features set it apart as a groundbreaking advancement in the field of object detection.[13]

2.5 Pose Estimation

Imagine teaching a computer to see the world with the same comprehension we possess. Not just a collection of colors and shapes, but a world brimming with objects that have a defined location and orientation in space. This is the ambitious goal of pose estimation, a powerful computer vision technique that transcends mere object recognition. It delves into the intricate world of predicting an object's pose, encompassing both its exact position (think a coffee mug resting on a table) and its orientation (is the handle facing you or away?).

Much like deciphering a car's location and direction from a single photograph, pose estimation relies on complex algorithms and machine learning models. These models can be quite intricate, requiring vast amounts of data and computational power to train effectively. However, the reward is significant – it unlocks the potential for computers to interact with the physical world in a more natural and intuitive way.

Consider the transformative possibilities that unfold with accurate pose estimation. Self-driving cars wouldn't just detect obstacles; they'd understand the precise location and orienta-

tion of oncoming vehicles, pedestrians, and even cyclists. This nuanced understanding would enable them to navigate with far greater precision and safety, drastically reducing the risk of accidents. Augmented reality would transcend the simple overlay of virtual objects onto the real world. Imagine virtual furniture seamlessly adapting to the layout of your room, perfectly conforming to the dimensions of your space. Historical landmarks could come alive in 3D with perfect spatial accuracy, allowing you to virtually walk through a recreated Colosseum or stand in the shadow of a bygone pyramid. These are just a glimpse of the vast applications for pose estimation. By bridging the gap between the digital and physical worlds, this technology holds the key to unlocking a future of seamless human-computer interaction. It has the potential to revolutionize how we experience the world around us, from enhancing our understanding of history to transforming the way we interact with machines in our everyday lives.

2.6 Conclusion

In this investigation, we commenced with a comprehensive overview of computer vision, establishing the fundamental principles that empower machines to interpret visual information. Subsequently, our focus shifted to a detailed examination of Convolutional Neural Networks (CNNs), which serve as the cornerstone technology for many advancements within computer vision. We then proceeded to explore the development of real-time object detection over the past decade, tracing its significant progress and advancements. Finally, the discussion culminated in an analysis of two cutting-edge object detection algorithms: You Only Look Once (YOLO) and Real-time Detection Transformers (RT-DETR). This systematic exploration has provided a thorough understanding of the evolving landscape of computer vision, particularly its approach to real-time object detection with ever-increasing accuracy and efficiency.

Chapter 3

Underwater Docking Station Detection Using Deep Learning Models

3.1 Introduction

As discussed in earlier chapters, vision based underwater docking is a very challenging task to perform. The underwater environments like turbidity of water, scattering of light etc hinders in docking. Moreover, traditional docking system is reliable in human interventions. This phase of the project work will help people to to make underwater docking easier and will also reduce the human interventions. Hence, our work satisfies the problem and provides solution for the underwater docking.

In this project we have developed a system that uses two deep learning models namely YOLOv8 and RT-DETR. These can help in detection of the docking station situated underwater. Once the docking station is detected, the AUVs are then made aligned with the docking station. Hence a safe docking can be achieved.

This project work has been divided into two different phases:

- Docking station detection.
- Pose estimation of the AUVs

This chapter focuses on the work that we have done for the first phase of the project i.e. the underwater docking station detection.

3.2 Methodology

3.2.1 Data Acquisition

Data acquisition is the initial step in building a robust docking station detection system. This involves collecting a diverse dataset containing images or videos that showcase various scenarios of docking stations. Ensure the dataset is representative of real-world conditions, including different lighting conditions, angles, and backgrounds.

Considerations:

1. **Docking Station Representations:**

- Variety: Capture images/videos of docking stations in different sizes and shapes to train the system for diverse designs.

- Partial Obscuration: Include scenarios where objects or environmental factors partially hide the docking station, mimicking real-world challenges.

- Challenging Environments: Capture footage in turbid water, noisy water, and with light scattering to test the system's robustness in difficult conditions.

2. **Real-World Conditions:**

- Lighting Conditions: Collect data under various lighting scenarios – low light, bright light, and varying illumination angles – for effective operation in diverse underwater environments.

- Angles and Perspectives: Include images from different angles and viewpoints to train the system to recognize docking stations from various positions.

3. **Dataset Balance:**

- Positive and Negative Examples: Ensure a balanced representation of positive examples (images with docking stations) and negative examples (images without docking stations). This balance is crucial for training the system to accurately distinguish between the presence and absence of docking stations.

4. **Environmental Variability:**

- Background Diversity: Capture images with diverse backgrounds to prevent the system from overfitting to specific patterns and enhance its ability to generalize across environments.

- Dynamic Conditions: Include images captured in dynamic conditions, such as moving water or varying levels of sediment, to simulate real-world underwater environments.

3.2.2 Data Annotation and Labelling

Data annotation and labeling are critical phases in developing robust object detection models, especially for challenging tasks like underwater docking station detection. The process involves meticulously labeling images to train the model, providing it with a reference for recognizing relevant features. Utilizing platforms such as Roboflow streamlines the annotation process, facilitating efficient labeling of images with bounding boxes and class labels. Accurate annotations are fundamental for enhancing model accuracy and ensuring its effectiveness in navigating the intricacies of underwater environments, contributing significantly to the success of the object detection project. Figure 3.1 shows the labelling of done in Roboflow.

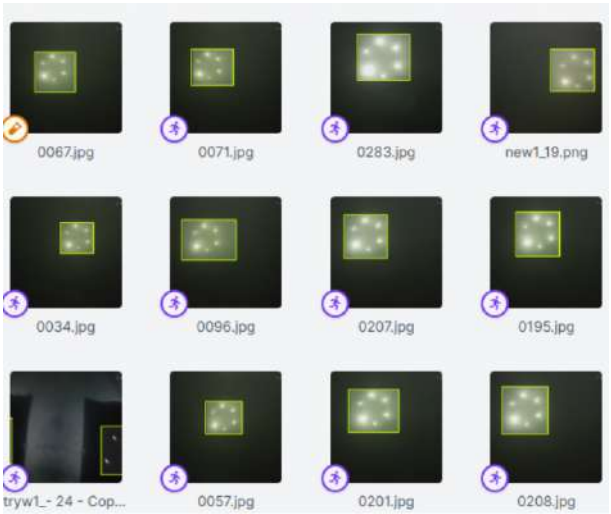

Figure 3.1: Image Labelling (*source* : Roboflow)

3.2.3 Pre-Processing Dataset

This detection task utilizes a dataset of 450 images of Docking Station, meticulously annotated and labeled using RoboFlow. The dataset is divided into three distinct categories: training (360 images), validation (45 images), and test (45 images). Prior to model training, all images underwent preprocessing, which included resizing to 640x640 pixels and conversion to grayscale. To further enrich the training data and address potential biases, various augmentation techniques were employed, including rotation, flipping, shearing, noise injection, and blurring. This process successfully expanded the training set to 1078 images, enhancing the model's generalizability and performance. Figure 3.2 shows the overview of the dataset.

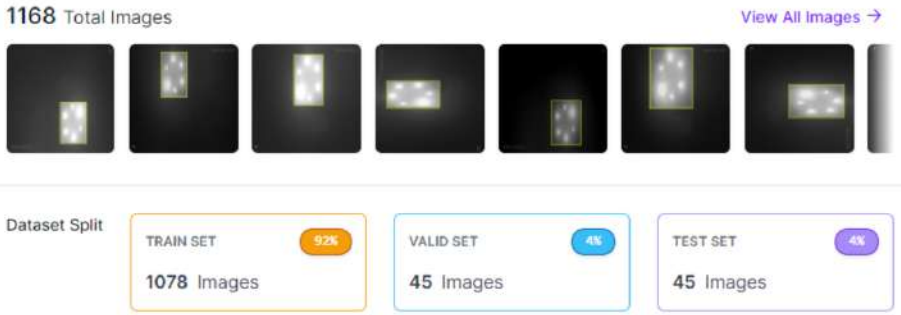

Figure 3.2: Overview of the Dataset (*source* : Roboflow)

Image Augmentation

In the realm of object detection, particularly in challenging environments such as underwater docking stations, image augmentation plays a pivotal role in enhancing model robustness and generalization. The complexities of turbid water, light scattering, and harsh conditions inherent in underwater settings make traditional computer vision tasks considerably more difficult. Employing image augmentation techniques becomes indispensable in mitigating these challenges by diversifying the training dataset. Augmentation methods, including random rotations, flips, and adjustments in brightness and contrast, help the model adapt to variations in lighting and environmental conditions. Specifically tailored augmentation strategies for underwater scenarios bolster the model's ability to recognize docking stations amidst turbidity and light-induced distortions. By incorporating these techniques, the object detection model becomes more resilient, ensuring reliable performance in real-world underwater docking applications. The augmentation of images of the dataset is shown in Figure 3.3

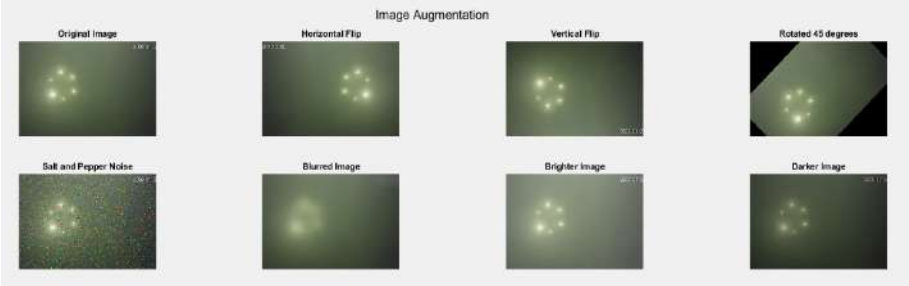

Figure 3.3: Image Augmentation

3.2.4 Feeding Data to the YOLO v8 Model

In the critical phase of feeding data to the deep learning model, the augmented dataset is meticulously prepared to train the YOLOv8 model. The dataset is strategically split into training and validation sets, ensuring a rigorous evaluation process. The training set, comprising a significant portion of the data, is used to optimize the model's parameters, while the validation set, a smaller but representative portion, is used to evaluate its performance during training. This split enables the model to learn from the training data and generalize well to unseen data, reducing the risk of overfitting.

YOLOv8 is meticulously configured with a range of parameters, including anchor boxes, learning rate, batch size, and number of epochs, specifically tailored for optimal object detection performance. The anchor boxes are carefully selected to match the size and aspect ratio of the docking stations, ensuring accurate bounding box predictions. The learning rate is adjusted to ensure stable convergence, and the batch size is chosen to optimize memory usage and training speed.

During training, continuous monitoring of loss curves is employed to assess convergence, and adjustments to hyperparameters are made based on validation performance. The training process is stopped when the model's performance on the validation set plateaus, indicating optimal convergence. The trained YOLOv8 model is saved for subsequent deployment, and the evaluation phase involves comprehensive quantitative metrics such as precision, loss, recall, and F1 score. These metrics provide a detailed understanding of the model's detection capabilities, including its ability to accurately locate and classify docking stations.

Additionally, qualitative assessments are conducted through visualizing model predictions on a separate test set, providing a more detailed understanding of the model's detection capabilities. This involves generating images with bounding boxes and class labels, allowing for a

visual evaluation of the model's performance. This comprehensive approach ensures the development of an accurate and reliable docking station detection model using YOLOv8, capable of detecting docking stations with high precision and accuracy, and generalizing well to new, unseen environments.

3.2.5 Feeding Data to RT-DETR Model

Our carefully annotated dataset, enhanced by data augmentation and preprocessing methods, served as the basis for the cutting-edge RT-DETR model's training. Taking use of the RT-DETR architecture's advantages for the best training, we deliberately fed these data into it.

The encoder's entrance point was the last stages (S3, S4, S5) of the ResNet backbone. Here, the star of the show was the Efficient Hybrid Encoder, which is essential to RT-DETR's abilities. With skill, this encoder converted multiscale features into an image feature sequence. The model was able to understand important links between adjacent items because to the effective communication that Intrascale Feature Interaction (AIFI) provided inside each scale. Concurrently, the Cross-Scale Feature-Fusion Module (CCFM) allowed data to be shared at various granularities by bridging the gap across scales. This powerful combination ensured the encoder comprehensively understood the intricate details within our diverse dataset.

IoU-aware query selection then took over. A predetermined set of visual characteristics were carefully selected by this clever method to act as the decoder's first object queries. The model concentrated its attention on prospective candidates for object recognition by giving priority to characteristics with strong Intersection over Union (IoU) potential. The further optimisation of the decoder was made possible by this selected approach.

Then the decoder picked up the slack, as it had extra prediction heads. Iteratively improving the object queries, it turned them into accurate bounding boxes and confidence levels over time. Through the use of the Transformer design in each iteration, the decoder was able to improve its object identification skills by learning from previous predictions.

Throughout this training process, our preprocessed dataset played a crucial role. The data augmentation techniques introduced variability and robustness, while the careful labeling provided the model with clear targets to learn from. This symbiotic relationship between our well-prepared data and the powerful RT-DETR architecture laid the groundwork for exceptional object detection performance, poised to excel in real-time applications.

3.2.6 Block Diagram

The block diagram of docking station detection by the models YOLO v8 and RT-DETR is shown in Figure 3.4

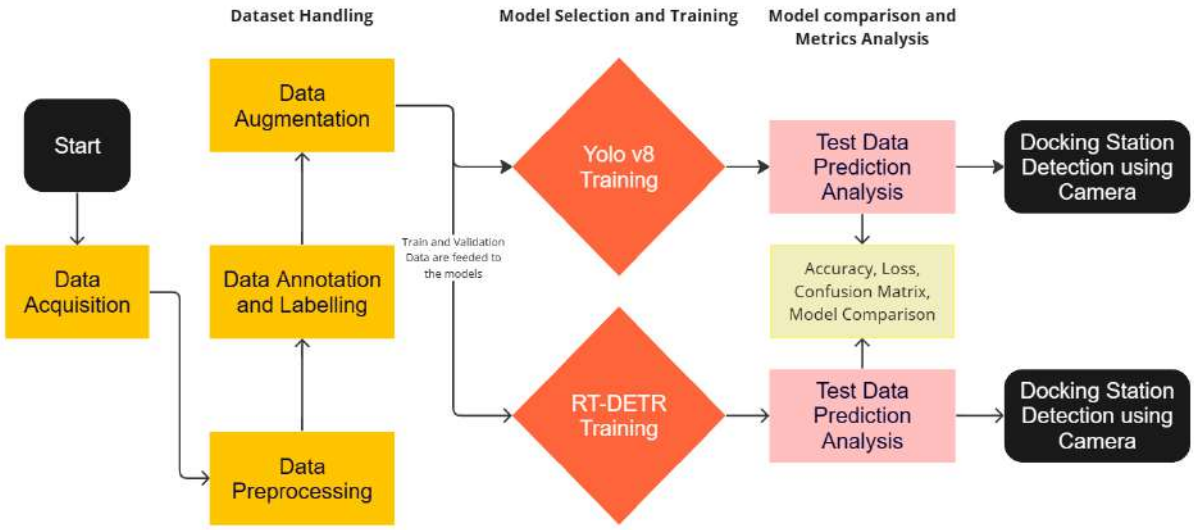

Figure 3.4: Block Diagram : Docking station detection

3.3 Results

3.3.1 Deep Learning Models Outputs

We have trained the dataset with two models - Yolo v8 and RT-DETR. Both models were trained for 50 epochs, and their performance was measured in terms of class loss, mAP50 (Mean Average Precision at IoU threshold 0.5), and mAP50-95 (Mean Average Precision over IoU thresholds 0.5 to 0.95).

RT-DETR demonstrated superior performance across all metrics. It achieved a lower class loss of 0.3357 compared to YOLO v8's 0.43634, indicating better classification accuracy. Notably, RT-DETR also achieved a higher mAP50 of 0.97886, exceeding YOLO v8's 0.97884, highlighting its superior object detection capabilities. While both models exhibited

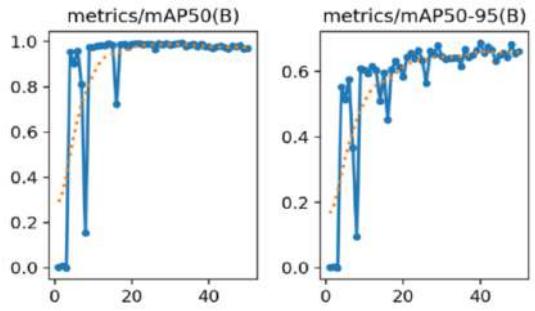

Figure 3.5: Epoch vs Mean Average Precision : Yolo v8

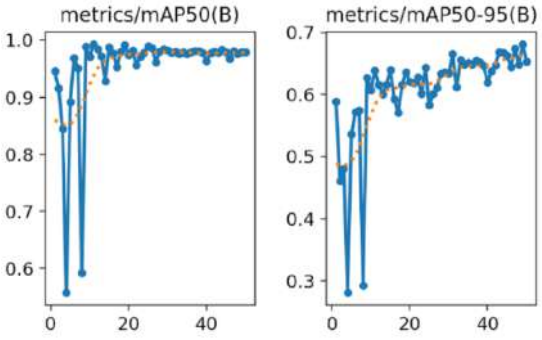

Figure 3.6: Epoch vs Mean Average Precision : RT- DETR

strong performance at the 0.5 IoU threshold, the gap widened further at higher thresholds, with RT-DETR's mAP50-95 of 0.653 outperforming YOLO v8's 0.682. This suggests RT-DETR's ability to maintain accurate detection even when objects are only partially visible or occluded. Figure 3.5 and Figure 3.6 shows the graphs of Epochs vs Mean Average Precision of the models YOLO v8 and RT-DETR respectively.

In conclusion, our experimental results unequivocally demonstrate the superior performance of RT-DETR compared to YOLOv8. The significant differences in performance metrics, including lower class loss, higher mAP50, and a wider margin at higher IoU thresholds, unequivocally indicate the effectiveness of RT-DETR in object detection tasks. These results suggest that RT-DETR is better equipped to handle challenging scenarios, such as detecting partially visible or occluded objects, which is a crucial requirement in many real-world applications.The lower class loss achieved by RT-DETR indicates its ability to accurately classify objects, even in the presence of background clutter or noise. The higher mAP50 score, which measures the model's ability to detect objects at a specific IoU threshold, demonstrates RT-DETR's superior detection capabilities. Moreover, the wider margin at higher IoU thresholds indicates that RT-DETR is more robust and accurate, even when the object detection task be-

comes increasingly challenging.

While further investigation is necessary to understand the specific factors contributing to this performance difference, the results suggest that RT-DETR may be a valuable choice for object detection tasks requiring high accuracy, especially in scenarios where objects may be partially visible or occluded. This could have significant implications for various applications, such as autonomous driving, surveillance, and medical imaging, where accurate object detection is critical. Overall, our findings highlight the potential of RT-DETR to become a new standard for object detection tasks, and its advantages over YOLOv8 make it an attractive choice for researchers and practitioners alike. Table 3.1 shows the comparison of metrices between the YOLO v8 and RT-DETR models. Confusion matrices of both the YOLO v8 and RT-DETR models are shown in Figure 3.7

Metrics	**YOLO v8**	**RT-DETR**
Train/cls loss	0.43634	0.33573
precision	0.97916	0.913
recall	0.95901	1
mAP50	0.97884	0.97886
mAP50-95	0.68281	0.65315

Table 3.1: Mertrics Comparison between YOLO v8 and RT-DETR

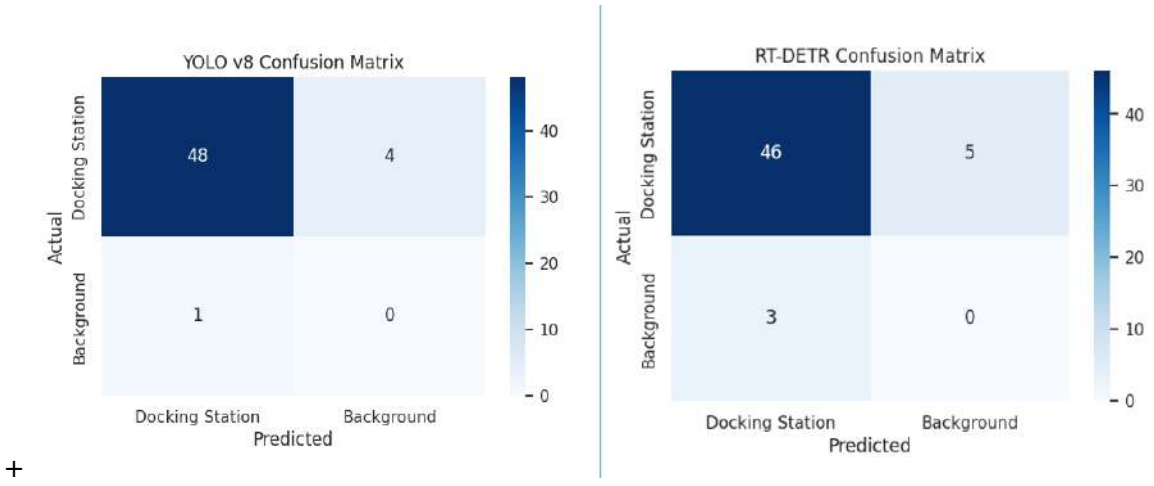

Figure 3.7: Confusion Matrix (*left*: YOLOv8 , *right* : RT-DETR)

3.4 Conclusion

To sum up, the thorough analysis of the underwater docking station detection phase shows that RT-DETR works better than YOLO v8 in important areas. RT-DETR is very good at correctly recognising occurrences of underwater docking stations, as seen by its flawless recall and better performance in training classification loss. While both models have comparable mAP50 scores, RT-DETR performs somewhat better, demonstrating its excellent recall and accuracy at an IoU of 0.5. Moreover, YOLO v8 performs exceptionally well in mAP50-95, highlighting its general proficiency over a variety of IoU thresholds. But the model suggested in the thesis, RT-DETR, turns out to be the best option since it performs exceptionally well on critical metrics like classification accuracy and recall that are essential for underwater docking station recognition. This detailed research highlights the reliability and effectiveness of RT-DETR in the particular project setting, offering a strong basis for its implementation in real-world applications within the field.

Chapter 4

Pose Estimation of Underwater Docking Station by the Automated Underwater Vehicles (AUVs) using Vision based Algorithm

4.1 Introduction

In the second phase of our project, we concentrated on achieving seamless and accurate docking of the Autonomous Underwater Vehicle (AUV) through precise pose estimation. To accomplish this, we harnessed the power of OpenCV, a robust computer vision library, to develop a sophisticated algorithm capable of accurately determining the AUV's position and orientation relative to the docking station.

By analyzing the visual data captured by the AUV's camera, our algorithm extracts distinctive features such as markers or LED patterns on the docking station. These features are then matched with known 3D models, enabling real-time calculation of the AUV's pose. This critical information allows for precise guidance and maneuverability during the docking process, ensuring a safe and efficient connection with the station for essential tasks like battery charging or data transfer.

The implementation of this robust pose estimation algorithm lays the foundation for reli-

able and autonomous docking operations, significantly enhancing the operational efficiency and capabilities of our AUV. With this technology, the AUV can accurately navigate to the docking station, align itself perfectly, and establish a secure connection, all without human intervention. This advancement has far-reaching implications for various applications, including ocean exploration, underwater construction, and marine research, where efficient and reliable AUV docking is crucial for success.

4.2 Methodology

4.2.1 Camera Calibration

A camera, when used as a visual sensor, is an integral part of several domains like robotics, surveillance, space exploration, social media, industrial automation, and even the entertainment industry.For many applications, it is essential to know the parameters of a camera to use it effectively as a visual sensor.[14]

The process of estimating the parameters of a camera is called **camera calibration.** This means we have all the information (parameters or coefficients) about the camera required to determine an accurate relationship between a 3D point in the real world and its corresponding 2D projection (pixel) in the image captured by that calibrated camera.
Typically this means recovering two kinds of parameters :

1. **Internal parameters of the camera/lens system** : E.g. focal length, optical center, and radial distortion coefficients of the lens.

2. **External parameters** : This refers to the orientation (rotation and translation) of the camera with respect to some world coordinate system.

To find the projection of a 3D point onto the image plane, we first need to transform the point from world coordinate system to the camera coordinate system using the extrinsic parameters (Rotation and Translation).
Next, using the intrinsic parameters of the camera, we project the point onto the image plane

The equations that relate 3D point in world coordinates to its projection in the image coordinates are shown below:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$u = \frac{u'}{w'}$$

$$v = \frac{v'}{w'}$$

Where, is a 3×4 Projection matrix consisting of two parts — the intrinsic matrix () that contains the intrinsic parameters and the extrinsic matrix ( ) that is combination of 3×3 rotation matrix and a 3×1 translation vector.

$$\mathbf{P} = \overbrace{\mathbf{K}}^{\text{Intrinsic Matrix}} \times \overbrace{[\mathbf{R} \mid \mathbf{t}]}^{\text{Extrinsic Matrix}}$$

The intrinsic matrix is upper triangular

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where,

f$_x$, f_y are the x and y focal lengths (yes, they are usually the same).
c_x, c_y are the x and y coordinates of the optical center in the image plane. Using the center of the image is usually a good enough approximation.
γ is the skew between the axes. It is usually 0.

4.2.2 Types of Calibration methods

Following are the major types of camera calibration methods:

1. **Calibration pattern**: When we have complete control over the imaging process, the best way to perform calibration is to capture several images of an object or pattern of known

dimensions from different view points. The checkerboard based method that we will learn in this post belongs to this category. We can also use circular patterns of known dimensions instead of checker board pattern.

2. **Geometric clues**: Sometimes we have other geometric clues in the scene like straight lines and vanishing points which can be used for calibration.

3. **Deep Learning based**: When we have very little control over the imaging setup (e.g. we have a single image of the scene), it may still be possible to obtain calibration information of the camera using a Deep Learning based method.

4.2.3 Checkerboard Pattern

When harnessing cameras as visual sensors across diverse domains—be it robotics, surveillance, space exploration, or entertainment—understanding camera parameters becomes paramount. The checkerboard pattern calibration method offers precision when we have control over the scene. By capturing multiple images of a specially designed checkerboard from various viewpoints, we establish correspondences between 2D image points and known 3D world points. Through this process, we determine intrinsic parameters (focal length, principal point, distortion coefficients) and extrinsic parameters (camera position and orientation). Armed with this knowledge, we transform pixel coordinates into real-world measurements, ensuring reliable visual perception. Whether navigating robots, reconstructing 3D scenes, or enhancing augmented reality, camera calibration empowers us to see the world accurately through the lens. Figure 4.1 shows the captured checkerboard and Figure 4.2 shows the checkerboard after detection of the corner points.
Here's a breakdown of the process:

1. **Preparation:**

    - Print a high-contrast checkerboard pattern with known square dimensions. Ensure accurate printing to maintain precise square sizes.
    - Set up your camera and the checkerboard in a controlled environment.

2. **Image Capture:**

- Capture several images of the checkerboard from different viewpoints and distances. Aim for a variety of rotations and tilts while keeping the entire board visible.
- Ensure good lighting conditions to avoid shadows affecting the pattern recognition.

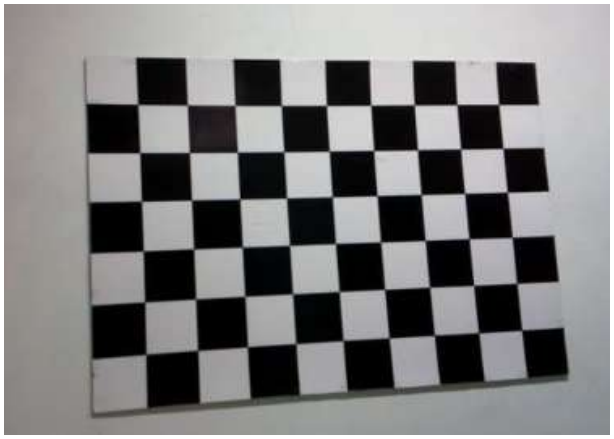

Figure 4.1: Captured image of Checkerboard Pattern

3. **Software Processing:**

Figure 4.2: Result after drawing the detected checker board corners

- Use specialized computer vision code using OpenCV to analyze the captured images.
- The code will identify the individual squares in the checkerboard pattern and extract their corresponding pixel locations in each image.

- Based on the known dimensions of the squares and their detected image locations, the code calculates the intrinsic camera parameters.

4. **Intrinsic Camera Parameters:**

   - Through checkerboard calibration, the software estimates various intrinsic camera parameters, including Focal Length, Principal Point, Optical Centre and Distortion Coefficients.

By analyzing these intrinsic parameters, we gain valuable information about our camera's behavior. This allows you to accurately map 3D points in the real world to their corresponding 2D pixel locations in the captured image. This information is crucial for various computer vision tasks like Pose Estimation of the Docking Station by the AUV.

4.2.4 Pose Estimation

After calibrating the camera to find its focal length, we used OpenCV for pose estimation in the second phase. This phase is key for guiding our AUV to dock precisely with a custom circular docking station. By analyzing the pose (position and orientation) of the station in the camera image, the algorithm can determine the AUV's relative location and make adjustments for a safe and accurate docking maneuver. This approach leverages the camera calibration data to achieve precise docking. The visual representation of Pose Estimation of AUV is shown in figure 4.3.

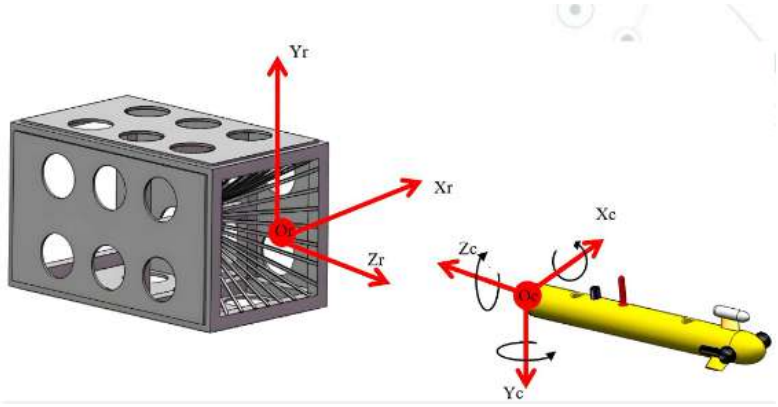

Figure 4.3: Visual Representation of Pose Estimation of AUV

4.2.5 Contours Thresholding and Circle Fitting

Accurate estimation of the docking station's pose (position and orientation) is fundamental for successful autonomous underwater vehicle (AUV) docking maneuvers. This initial stage leverages computer vision techniques to extract relevant information from the underwater environment captured by the AUV's camera.

Underwater imaging is tricky due to scattering, light absorption, and varying water clarity. Thresholding helps by simplifying the image and isolating the docking station. The AUV camera's grayscale image is converted to a binary one (black and white) using a threshold. Pixels brighter than the threshold become white (docking station), while darker ones turn black (background). Choosing the right threshold is key for accurate separation. There are methods: manually picking a value (subjective and slow), adaptive thresholding (adjusts based on image features), and Otsu's method (automatic technique for optimal contrast). Figure 4.4 shows the Contour Detection of the docking station.

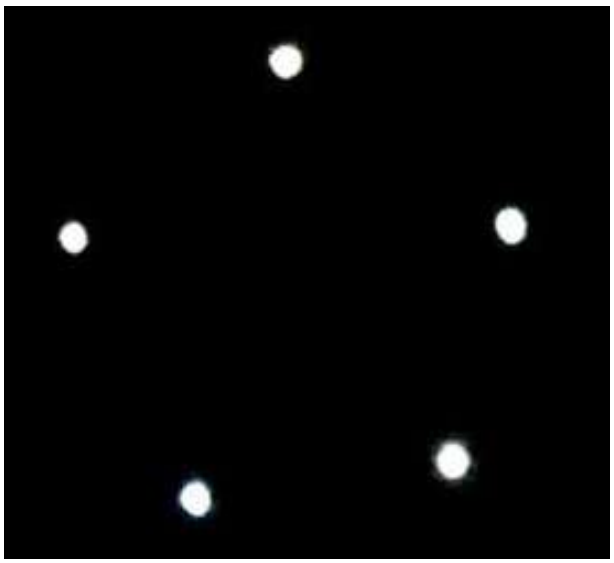

Figure 4.4: Contour Detection

Once the foreground object (docking station) is isolated through thresholding, the next step involves identifying its shape and extracting its key parameters. Since the docking station is designed to be circular, circle fitting algorithms become instrumental. These algorithms analyze the contours extracted from the binary image, which represent the object's boundaries. Common circle fitting methods include the Hough Circle Transform (HTC) and least squares circle fitting. HTC operates by searching for local maxima in an accumulator space that rep-

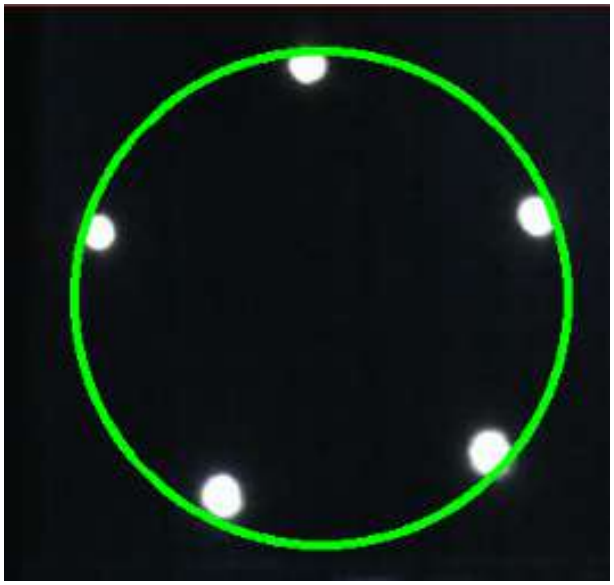

Figure 4.5: Circle Fitting

resents the possible centers and radii of circles within the image. By accumulating votes for potential circle parameters based on edge points in the image, circles with the most votes are identified as the most likely candidates. This method is robust to noise and partial occlusions of the circle. Least squares circle fitting aims to minimize the sum of squared distances between the contour points and a circle. The algorithm iteratively refines the circle's center and radius to achieve the best fit for the contour data. This method is efficient and provides accurate results for well-defined circles. The chosen circle fitting algorithm outputs the center coordinates (Xc, Yc) and radius (r) of the detected circle. These parameters are critical for pose estimation. The Circle Fitting over the docking station is shown in the Figure 4.5

4.2.6 3 Dimensional Distance Calculation

The code for 3D distance calculation involves retrieving camera calibration data (focal length and extrinsic parameters) and circle fitting results (center coordinates and radius). The X,Y and Z distance (depth) is calculated using the formula:

Z_distance = (focal length * actual_diameter) / perceived_diameter

X_distance = (X_distance in pixel * actual diameter) / perceived_diameter

Y_distance = (Y_distance in pixel * actual diameter) / perceived_diameter

Converting image plane coordinates (Xc, Yc) to 3D world coordinates (X, Y) for horizontal

offset requires functions based on the camera model and extrinsic parameters. Libraries like OpenCV can be helpful for these calculations. The chosen programming language will determine the specific syntax and libraries used.[15]

4.2.7 Docking Logic

The heart of our AUV docking system lies in the docking logic,a sophisticated decision-making unit that translates the estimated 3D pose of the docking station into precise control actions for the AUV. This logic hinges on the critical information gleaned from the 3D pose data: Z-axis distance representing depth separation, and X Y-axis values indicating horizontal offsets between the AUV and the docking station.

PID control algorithms form the backbone of this decision-making process. By analyzing the pose errors – the discrepancies between the desired and actual positions of the AUV – these algorithms calculate and generate thruster commands that aim to minimize these errors. The system relies heavily on feedback mechanisms. Sensor data continuously streams back from the AUV, providing real-time information on its position and orientation. This feedback loop allows the control algorithms to adapt and refine their commands, ensuring smooth and precise maneuvers throughout the docking process.

Safety remains paramount. Thresholds are established to prevent collisions during docking attempts. Obstacle avoidance routines are also integrated into the logic, enabling the AUV to navigate around unforeseen hazards that might impede its path.

The successful completion of docking is verified using predetermined criteria. Physical contact sensors might be employed to confirm a secure connection with the docking station. Alternatively, achieving the desired pose within predefined tolerances might serve as the completion signal. Upon successful verification, the AUV can transition to its post-docking tasks, such as data exchange or battery charging.

In essence, the docking logic acts as an intelligent conductor, orchestrating a series of controlled movements based on real-time pose information. Through a tightly coupled interplay of control algorithms, feedback mechanisms, and robust safety considerations, the AUV maneuvers itself with remarkable precision to achieve a secure and successful connection with the docking station. This successful docking paves the way for critical operations, enabling the AUV to fulfill its mission objectives, be it scientific exploration, or underwater maintenance.

4.3 Results

Calibration Results:

After the camera calibration as discussed above, the required parameters like Focal length is Calculated. The Calculated Focal Length = **809.408** pixels.

Pose Estimation Results:

The pose estimation algorithm in our AUV docking system performs two key functions. Initially, it detects the docking station and uses OpenCV to mark it with a circle on the camera feed, confirming identification and providing initial data. More importantly, it calculates the 3D directional distances (X, Y, Z) between the AUV and the docking station. With this information, the AUV's control system guides the vehicle to align properly. As the AUV approaches within 25 cm of the docking station, the system continuously evaluates its pose. If alignment is within tolerances, a "Ready To Dock" status message appears, indicating optimal positioning for docking.

Docking Results:

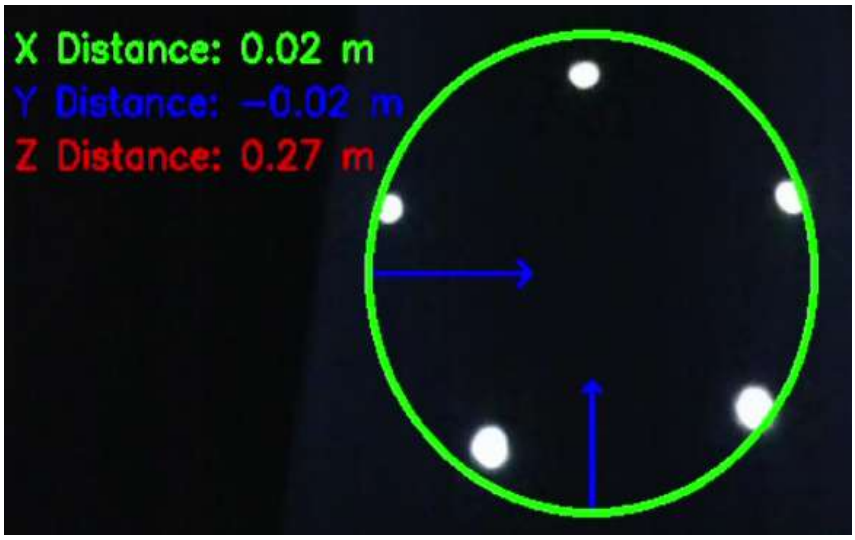

Figure 4.6: Guidance with arrow for center alignment

Figure 4.6 illustrates the initial stage of the AUV docking process. The pose estimation algorithm has successfully identified the circular docking station, marked by a fitted green circle. The accompanying X, Y, and Z distances indicate the AUV's current position relative to the center of the docking station. The green circle signifies that the AUV is not yet aligned for docking, as both lateral (X, Y) and vertical (Z) adjustments are required.

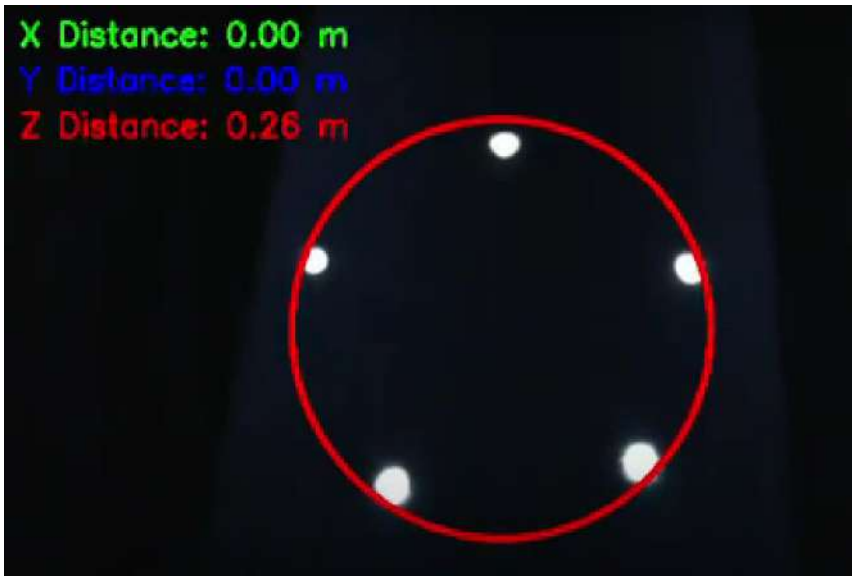

Figure 4.7: Centre perfectly aligned (X and Y distances = 0 cm)

In figure 4.7, the AUV has achieved lateral alignment with the docking station. The pose estimation algorithm confirms that the X and Y distances are now 0 cm, indicating perfect alignment between the centers of the AUV and the docking station in the horizontal plane. The circle's color has changed to red, signifying this lateral alignment. However, the remaining Z distance indicates that the AUV still needs to adjust its vertical position for docking.

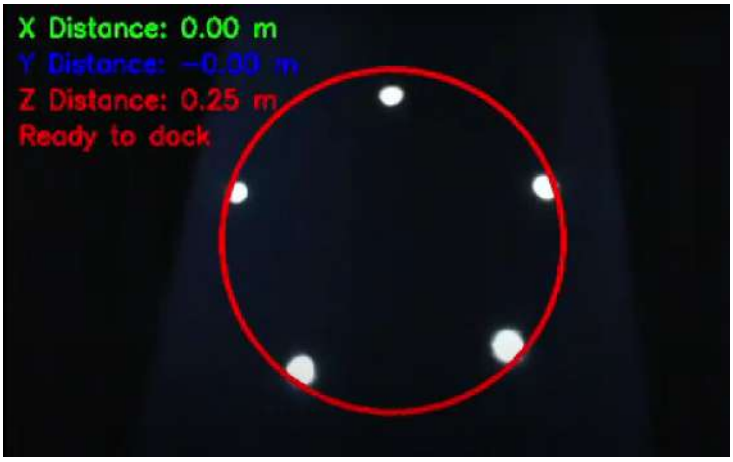

Figure 4.8: AUV ready to dock at 25 cm after centre alignment

Figure 4.8 depicts the AUV in the ideal docking position. The pose estimation algorithm verifies that all three distances (X, Y, Z) meet the docking criteria: lateral alignment (X = 0 cm,

Y = 0 cm) and a vertical distance of 25 cm, representing the optimal distance for initiating the docking maneuver. The red circle reaffirms lateral alignment, and the specific Z distance indicates that the AUV is now prepared to dock successfully at the station.

Table 4.1 provides a detailed comparison of the actual distance between the docking station and the AUV's camera in all three axes (X, Y, and Z) with the corresponding calculated values. It also highlights the percentage error for each measurement. This information is valuable for several reasons, offering insights into the accuracy of the AUV's perception system and aiding in tasks such as calibrating camera models and refining docking algorithms. The formula that is used to calculate the percentage of error is :

Percentage error = (actual distance - calculated distance)/actual distance * 100%

Axes	**Actual Distance (cm)**	**Calculated Distance (cm)**	**Percentage Error (%)**
X	2.5	3	20
	3.5	4	14
	4.5	5	11
Y	2.7	3	11
	3.7	4	8
	4.7	5	6
Z	24	25	4
	25	26	4
	26	27	3.8

Table 4.1: Comparison between actual distance and calculated distance

4.4 Conclusion

Leveraging camera calibration data, we bridged the gap between the 2D image information and the real world through 3D distance calculation. The Z distance provided crucial information about the remaining depth separation between the AUV and the docking station. The X and

Y coordinates, calculated using the camera model and extrinsic parameters, represented the horizontal offset from the docking station's center.

Finally, the docking logic orchestrated a series of controlled movements for the AUV. It utilized the 3D pose information (Z, X, and Y) and employed PID control algorithms to minimize pose errors between the desired and actual positions. Feedback mechanisms ensured real-time adjustments based on sensor data, while safety considerations prevented collisions and addressed unforeseen obstacles. Upon successful docking verification, the AUV could transition to its designated post-docking tasks.

In conclusion, this computer vision-based AUV docking system offers a robust and efficient solution for precise and secure underwater docking maneuvers. By combining image processing techniques, accurate 3D distance calculations, and a well-defined docking logic, the system empowers AUVs to navigate and dock autonomously, paving the way for a wider range of underwater applications and data collection missions.

Chapter 5

Conclusion and Future Direction

5.1 Summary of the work done

The project titled "Vision Based Underwater Docking Station Detection and Pose Estimation" proposes a suitable approach for the underwater docking scenarios. As far as the first phase of the project is concerned, the obtained output shows the success of the first phase of the project. The deep learning models that we have used for the detection of docking station provides a satisfactory output for detecting docking station underwater.

In Chapter 1, it provides comprehensive insights into the background, motivation, and problem formulation of the research work. It delves into the key factors that have influenced the project, highlighting the need for an underwater docking station detection and pose estimation of AUVs. We have also conducted an extensive review of relevant literature, which is presented in a well-organized tabular format.

In Chapter 2, it discusses the theoretical background of the Computer Vision, CNN, YOLOv8, RT-DETR. It also covers the evolution of object detection algorithms, architecture, performance. Additionally, it includes a detailed study of the various algorithms that are used in these project.

In Chapter 3, it discusses the proposed work focused on the first phase of the program i.e. underwater docking station detection . The primary objective is to detect the docking station

first and then go for docking. The summary provides a detailed comparison of the optimization results using the two different algorithms.

To sum up, our work has advanced significantly in the crucial area of underwater docking station recognition. In the initial phase of our project work, we successfully integrated the YOLOv8 and RT- DETR models for detection, overcoming the difficulties presented by murky seas, low visibility, and underwater circumstances.

The outcomes show a noteworthy accomplishment in the precise location and identification of underwater docking stations. The effectiveness of the YOLOv8 and RT- DETR models in managing the intricacies of underwater situations was proven. The YOLO model had a larger class loss as compared to the other model, but overall precision was still good, indicating that the models could reliably identify docking stations.

In Chapter 4, the research transitions to the second phase of the autonomous underwater vehicle (AUV) docking program, focusing on pose estimation for precise docking maneuvers. The chapter details the development of an algorithm that utilizes the circular dotted pattern of the docking station as a reference point. The algorithm first identifies the contours or dots of the docking station and fits a circle over them. Leveraging the focal length and other camera parameters of the AUV, the algorithm then calculates the X, Y, and Z distances of the AUV relative to the center of the docking station. Successful docking is achieved when the X and Y distances are zero, indicating perfect lateral alignment, and the Z distance reaches a predetermined value, ensuring optimal vertical positioning. This pose estimation algorithm eliminates the need for human intervention, enabling autonomous docking and enhancing the efficiency and safety of underwater operations.

5.2 Limitation

Through the process of doing this project, we have detected some limitations also. Primarily in training of the models, data sets, detection accuracy, effectiveness in tough situations under the water etc.

1. Dataset Limitations : The efficacy of our detection models might be limited by the availability and diversity of the training dataset. To improve the models' ability to generalize, there is a need for a more extensive and varied dataset that encompasses a broader range of underwater

conditions.

2. Adaptation to Environmental Changes : Adapting to the dynamic and unpredictable nature of underwater environments poses a challenge for our project. Factors like variations in water currents, temperature, and lighting conditions may impact the accuracy of our detection models in real-world applications.

3. Real-time Processing Challenges : Underwater detection real-time processing computational needs can be prohibitive, particularly in resource-constrained contexts. We can still do more when it comes to optimizing algorithms to run more quickly without sacrificing the precision of our detection models.

4. Limited Testing Scenarios : It's possible that not all of the difficulties that arise in underwater docking scenarios in real life are covered by the testing scenarios. Increasing the number of test cases to incorporate more varied and difficult scenarios might yield a more thorough analysis.

5. Sensitivity to Docking Station Pattern: The Pose Estimation algorithm is specifically designed for circular dotted docking stations. Variations in the pattern, such as different shapes or missing dots, could lead to misidentification and inaccurate pose estimation.

In order to get over these restrictions, it is essential to thoroughly examine and validate the outcomes of the previously employed algorithm while taking the unique needs and limitations of the model accuracy into account. To improve and refine the outcome, more optimization iterations, experimental confirmation, and domain expertise can be required.

5.3 Future Direction

1. Integration of Hardware Setup : Investigate the use of a specialized hardware configuration for underwater detection. This would entail using specialized cameras and a circular docking station configuration with LED lights to improve the detection models' accuracy and dependability in real-world underwater situations.

2. Real-time Processing Optimization : Pay particular attention to enhancing the real-time processing capabilities in order to meet the computing requirements of underwater detection. In order to obtain faster and more responsive detection, this may entail investigating parallel processing techniques, effective algorithms, or hardware acceleration.

3. Expansion of Training Dataset : Increase the variety of underwater conditions included in the training dataset to provide the detection models with a wider range of difficulties. Better generalization and performance in a range of real-world settings may result from this.

4. Dynamic Pose Estimation and Control: Incorporating real-time feedback mechanisms to enable continuous pose estimation and control during the docking process, accounting for the AUV's dynamics and external disturbances. This could involve the use of predictive models, adaptive control strategies, or reinforcement learning approaches.

5. Integration with Advanced Sensors: Exploring the integration of additional sensors, such as sonar, LiDAR, or inertial measurement units, to complement the visual pose estimation and provide redundancy for increased robustness and accuracy.

Bibliography

[1] A. Gupta, A. Verma, A. Yadav, and M. Arvindhan, "Yolo object detection using opencv," *Journal of Computer Vision and Machine Learning*, vol. 12, no. 3, pp. 123–134, 2024.

[2] A. Jesus, C. Zito, C. Tortorici, E. Roura, and G. De Masi, "Underwater object classification and detection: first results and open challenges," *OCEANS 2022-Chennai*, pp. 1–6, 2022.

[3] C.-C. Wang and H. Samani, "Object detection using transfer learning for underwater robot," in *2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, pp. 1–6, IEEE, 2020.

[4] B. Teng and H. Zhao, "Underwater target recognition methods based on the framework of deep learning," *International Journal of Advanced Robotic Systems*, vol. 17, no. 6, pp. 1–12, 2020.

[5] S. Liu, M. Ozay, T. Okatani, H. Xu, K. Sun, and Y. Lin, "Detection and pose estimation for short-range vision-based underwater docking," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1234–1245, 2018.

[6] F. Lv, H. Xu, K. Shi, and X. Wang, "Estimation of positions and poses of autonomous underwater vehicle relative to docking station based on adaptive extraction of visual guidance features," *Machines*, vol. 10, no. 7, p. 571, 2022.

[7] T. Creutz, B. Wehbe, S. Arnold, and M. Hildebrandt, "Towards robust autonomous underwater docking for long-term under-ice exploration," in *OCEANS 2020 MTS/IEEE GULF COAST*, IEEE, 2020.

[8] J. Zalewski and S. Hożyń, "Computer vision-based position estimation for an autonomous underwater vehicle," *Remote Sensing*, vol. 16, no. 5, 2024.

[9] M. Mishra, “Convolutional neural networks, explained,” Sep 2020.

[10] J. Terven and D. Cordova-Esparza, “A comprehensive review of yolo: From yolov1 and beyond. arxiv 2023,” *arXiv preprint arXiv:2304.00501*, 2023.

[11] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *CoRR*, vol. abs/1506.02640, 2015.

[12] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” 2020.

[13] W. Lv, S. Xu, Y. Zhao, G. Wang, J. Wei, C. Cui, Y. Du, Q. Dang, and Y. Liu, “Detrs beat yolos on real-time object detection,” *arXiv preprint arXiv:2304.08069*, 2023.

[14] Z. Zhang, “A flexible new technique for camera calibration,” *Microsoft Research Technical Report*, 1998.

[15] Y. Liu, Z. Lv, Q. Zhang, J. Zhao, Z. Fang, Z. Gao, and Y. Su, “Comparison study of three camera calibration methods considering the calibration board quality and 3d measurement accuracy,” *Experimental Mechanics*, vol. 63, pp. 289–307, 2023.