ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

INTERNATIONAL HELLENIC UNIVERSITY

DEPARTMENT OF INFORMATION AND ELECTRONIC ENGINEERING

DIPLOMA THESIS

«Real-Time Vision-Based Marine Debris Detection on Embedded Autonomous Surface Vehicles»

Students:
Moustafa BEKIR

Supervisor:
George Kokkonis
Assistant Professor

IAN 2026

Title of Dissertation Real-Time Vision-Based Marine Debris Detection on Embedded Autonomous Surface Vehicles

Code of Dissertation 25000

Student's full name Moustafa BEKIR

Supervisor's full name George Kokkonis

Date of undertaking

Date of completion

I hereby affirm the authorship of this paper as well as the acknowledgement and credit of whichever assistance I received in its composition. I have, furthermore, noted the various sources from which I extracted data, ideas, visual or written material, in paraphrase or exact quotation. Moreover, I affirm the exclusive composition of this paper by myself only, for the purpose of it being a dissertation, in the Department of Information and Electronic Engineering of the I.H.U.

The approval of this dissertation by the Department of Information and Electronic Engineering of the International Hellenic University, does not necessarily entail the adoption of the author's views, on behalf of the Department.

Prolog

This diploma thesis was conducted at the Department of Information and Electronic Engineering of the International Hellenic University, as part of the requirements for the completion of the undergraduate curriculum. The subject of the thesis, the development of an autonomous system for marine debris detection, was selected out of a desire to bridge the gap between theoretical computer science and practical environmental engineering.

The motivation for this research stems from the growing global challenge of marine pollution. While significant progress has been made in deep learning, the application of these technologies in resource-constrained, real-world aquatic environments remains a complex engineering challenge. This work represents an effort to design a system that is not only algorithmically effective but also physically deployable and ecologically responsible.

The journey of designing, building, and programming the autonomous surface vehicle presented in this volume was both demanding and rewarding. It required integrating knowledge from diverse fields, including mechanical design, embedded systems, and computer vision. This thesis reflects the culmination of the knowledge and skills acquired during my studies and serves as a foundation for future research in the field of environmental robotics.

.

«Real-Time Vision-Based Marine Debris Detection on Embedded Autonomous Surface Vehicles»

«Moustafa BEKIR»

Abstract

EN

The proliferation of marine debris presents a critical environmental challenge, particularly in shallow harbor and riverine environments where conventional cleanup vessels and crewed surveys are ineffective or impractical. This thesis presents the design, development, and experimental validation of a low-cost Autonomous Surface Vehicle (ASV) for real-time vision-based debris detection in near-surface aquatic environments.

The proposed system is built on a custom-fabricated catamaran platform with differential thrust propulsion and an embedded computing architecture based on the Raspberry Pi 5, enabling fully onboard perception and navigation. To address the visual degradation inherent to aquatic imaging—such as turbidity, light attenuation, and surface reflections—deep learning–based object detection models were trained and evaluated using the TrashCan 1.0 dataset.

A comparative experimental study was conducted on lightweight object detection architectures suitable for embedded deployment, including YOLOv5n, YOLOv8n, YOLO11n, and the transformer-based **RT-DETR**. Model performance was evaluated with respect to detection accuracy and real-time inference feasibility on embedded hardware. **Furthermore, a novel "Bio-Safety" taxonomy was implemented, explicitly training the system to distinguish between anthropogenic debris and marine life to minimize ecological disruption.**

The results demonstrate that modern nano-scale convolutional detectors can achieve a favorable balance between accuracy and computational efficiency when applied to aquatic debris detection. In particular, YOLO11n exhibited the most robust performance under embedded constraints, supporting real-time operation without external computation. These findings confirm the feasibility of low-power, perception-driven ASVs as scalable tools for autonomous monitoring of aquatic pollution.

Keywords: Autonomous Surface Vehicle, Marine Debris Detection, Object Detection, YOLO11, RT-DETR, Edge Computing, Bio-Safety.

EL

Η εξάπλωση των θαλάσσιων απορριμμάτων αποτελεί μια κρίσιμη περιβαλλοντική πρόκληση, ιδιαίτερα σε ρηχά λιμεναρχεία και ποτάμια περιβάλλοντα όπου τα συμβατικά σκάφη καθαρισμού είναι αναποτελεσματικά. Η παρούσα διπλωματική εργασία παρουσιάζει τον σχεδιασμό, την ανάπτυξη και την πειραματική επικύρωση ενός χαμηλού κόστους Αυτόνομου Σκάφους Επιφανείας (ASV) για τον εντοπισμό απορριμμάτων σε πραγματικό χρόνο.

Το προτεινόμενο σύστημα βασίζεται σε μια κατασκευή τύπου καταμαράν και μια αρχιτεκτονική ενσωματωμένων συστημάτων με βάση το Raspberry Pi 5, επιτρέποντας πλήρη επεξεργασία δεδομένων επί του σκάφους. Για την αντιμετώπιση των οπτικών προκλήσεων του υδάτινου περιβάλλοντος—όπως η θολότητα και οι αντανακλάσεις—μοντέλα βαθιάς μάθησης εκπαιδεύτηκαν και αξιολογήθηκαν χρησιμοποιώντας το σύνολο δεδομένων TrashCan 1.0.

Πραγματοποιήθηκε συγκριτική μελέτη σε ελαφριές αρχιτεκτονικές ανίχνευσης αντικειμένων, συμπεριλαμβανομένων των YOLOv5n, YOLOv8n, YOLO11n και του μοντέλου RT-DETR. Επιπλέον, εφαρμόστηκε μια καινοτόμος ταξινομία «Βιο-Ασφάλειας» (Bio-Safety), εκπαιδεύοντας το σύστημα να διακρίνει τα ανθρωπογενή απορρίμματα από τη θαλάσσια ζωή για την ελαχιστοποίηση της οικολογικής διαταραχής.

Τα αποτελέσματα αποδεικνύουν ότι οι σύγχρονοι ανιχνευτές νανο-κλίμακας (nano-scale) επιτυγχάνουν ευνοϊκή ισορροπία μεταξύ ακρίβειας και υπολογιστικής απόδοσης. Συγκεκριμένα, το YOLO11n επέδειξε την πιο ισχυρή απόδοση υπό περιορισμούς υλικού, υποστηρίζοντας λειτουργία σε πραγματικό χρόνο. Τα ευρήματα αυτά επιβεβαιώνουν τη βιωσιμότητα των αυτόνομων σκαφών ως κλιμακώσιμα εργαλεία για την παρακολούθηση της υδάτινης ρύπανσης.

Λέξεις-Κλειδιά: Αυτόνομο Σκάφος Επιφανείας, Ανίχνευση Θαλάσσιων Απορριμμάτων, YOLO11, RT-DETR, Edge Computing, Βιο-Ασφάλεια.

.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, **Assistant Professor George Kokkonis**, for his guidance, technical expertise, and support throughout the development of this thesis.

I also thank my family for their encouragement and patience during my studies

.

.

Content

List of Figures

List of Tables

Abbreviations

AI Artificial Intelligence

ASV Autonomous Surface Vehicle

CAD Computer-Aided Design

CNN Convolutional Neural Network

COCO Common Objects in Context

CSI Camera Serial Interface

DETR Detection Transformer

ESC Electronic Speed Controller

FDM Fused Deposition Modeling

FN False Negative

FP False Positive

FPR False Positive Rate

FPS Frames Per Second

FPV First Person View

GNSS Global Navigation Satellite System

GPS Global Positioning System

I2C Inter-Integrated Circuit

IMU Inertial Measurement Unit

IoU Intersection over Union

mAP Mean Average Precision

PLA Polylactic Acid

RAM Random Access Memory

RGB Red, Green, Blue

ROV Remotely Operated Vehicle

RT-DETR Real-Time Detection Transformer

SGD Stochastic Gradient Descent

SSD Single Shot MultiBox Detector

TP True Positive

UART Universal Asynchronous Receiver-Transmitter

YOLO You Only Look Once

Chapter 1: Introduction

1.1 Background and Motivation

Marine pollution has emerged as a persistent and complex environmental challenge, with plastic debris increasingly accumulating in rivers, estuaries, and near-shore environments [1]. These regions act as primary transport pathways through which waste enters larger marine ecosystems, yet they are often inaccessible to large cleanup vessels and economically impractical to monitor using conventional survey methods. As a result, shallow and confined waterways remain under-monitored despite their disproportionate contribution to marine debris propagation.

Traditional approaches to debris monitoring, including manual shoreline surveys, diver-based inspections, and remotely operated vehicles (ROVs), suffer from fundamental limitations [2]. Manual surveys are labor-intensive and lack temporal continuity, while ROV-based operations require tethered deployment, specialized vessels, and skilled operators, significantly increasing cost and logistical complexity. Moreover, these approaches are poorly suited for persistent monitoring of surface and near-surface debris, where visibility conditions and dynamic water motion complicate sustained observation.

Autonomous Surface Vehicles (ASVs) provide a practical alternative by enabling repeated, unmanned surveys of shallow and hazardous aquatic environments [3]. Operating directly at the air–water interface, ASVs are well positioned to observe floating and partially submerged debris using optical sensors while maintaining lower operational costs than crewed platforms. However, enabling reliable autonomy on such platforms introduces a set of tightly coupled engineering challenges. Vision-based perception systems deployed on ASVs must operate under visually degraded conditions caused by surface reflections, turbidity, refraction, and dynamic illumination, while simultaneously meeting strict constraints on onboard computation, power consumption, and real-time responsiveness.

Recent advances in deep learning–based object detection have demonstrated impressive performance on large-scale terrestrial datasets. Nevertheless, models achieving high benchmark accuracy often rely on substantial computational resources and are trained on imagery that does not reflect aquatic visual conditions. When deployed on embedded hardware aboard ASVs, these models frequently experience degraded performance due to domain mismatch and insufficient inference speed. This creates a critical gap between algorithmic performance reported in the literature and practical feasibility in real-world ASV deployments.

The motivation for this research arises from this gap. Rather than proposing new detection architectures, this thesis focuses on the systematic evaluation and deployment-aware selection of modern object detection models for ASV-based debris monitoring. Emphasis is placed on identifying lightweight architectures capable of sustaining real-time inference on embedded hardware while remaining robust to aquatic visual degradation.

Crucially, this work does not aim to deliver a fully autonomous or production-ready ASV system, but instead seeks to demonstrate the feasibility and operational capability of deploying real-time, vision-based debris detection on a small-scale ASV platform under realistic environmental and computational constraints.

By integrating platform design, navigation, and perception into a unified system and validating performance through field experiments, this work aims to establish a practical foundation that can inform and guide future research toward more robust and fully autonomous ASV deployments in near-surface environments.

1.2 Objectives of the Study

The primary objective of this study is to design, implement, and experimentally evaluate a low-cost Autonomous Surface Vehicle (ASV) capable of real-time, vision-based detection of surface and near-surface marine debris under embedded deployment constraints. The work emphasizes practical system integration and deployment-aware evaluation rather than the development of novel navigation or perception algorithms.

To achieve this objective, the study is structured around the following specific goals:

1. Design and Fabrication of an Embedded ASV Platform
Develop a lightweight and stable ASV platform suitable for operation in shallow and confined waterways, with mechanical and electrical design choices informed by the requirements of onboard vision-based perception and low-power operation.

2. Integration of Autonomous Navigation and Control Using a Marine Autopilot

Integrate a Pixhawk-based autopilot system to provide low-level navigation, stabilization, and control functions, including GPS-based localization, inertial sensing, and closed-loop motion control. The focus is placed on reliable system integration and validation of autonomous operation in support of perception-driven tasks, rather than on the development of new navigation algorithms.

3. Implementation and Comparison of Vision-Based Object Detection Models
Implement multiple representative modern object detection architectures, including lightweight convolutional models and representative transformer-based detectors, for the task of marine debris detection using a downward-facing camera. This objective explicitly includes comparative evaluation to identify models suitable for real-time embedded deployment.

4. Embedded Performance Evaluation Under Deployment Constraints
Evaluate detection models using metrics that reflect real-world ASV operation, including detection accuracy (mAP@50), inference latency on a Raspberry Pi 5, and qualitative detection stability under aquatic visual degradation. This objective emphasizes the trade-off between perception performance and computational feasibility on embedded hardware.

5. Experimental Validation in Representative Field Conditions
Validate the integrated ASV system through controlled and limited real-world experiments, assessing the feasibility of sustained autonomous operation and onboard perception in dynamic surface environments.
Collectively, these objectives ensure that the thesis remains tightly aligned with practical ASV deployment scenarios. By prioritizing system integration, embedded evaluation, and empirical

comparison of existing detection architectures, the study aims to provide actionable insights into the feasibility of real-time vision-based debris detection on autonomous surface platforms.

1.3 Scope of the Thesis

This thesis focuses on the design, implementation, and experimental validation of an Autonomous Surface Vehicle (ASV) for vision-based detection **of surface and near-surface marine debris** in shallow and confined aquatic environments. The scope is intentionally defined to emphasize practical system integration, embedded deployment, and empirical performance evaluation under realistic operational constraints.
The work encompasses the following core aspects:

1. **Hardware Design and System Integration**
   The study includes the mechanical and electrical design of a low-cost ASV platform, with emphasis on stability, payload integration, and suitability for onboard optical sensing. Hardware integration covers propulsion, power distribution, sensor mounting, and embedded computation using a Raspberry Pi 5, selected to reflect realistic constraints of small autonomous platforms.
2. **Navigation and Control Integration**
   The thesis addresses the integration and validation of autonomous navigation and control capabilities using a Pixhawk-class marine autopilot. This includes GPS-based localization, inertial sensing, and closed-loop motion control required to support perception-driven operation. The work does not aim to develop or optimize novel navigation algorithms, but rather to ensure reliable system-level operation in support of onboard perception tasks.
3. **Vision-Based Perception and Object Detection**
   The implementation and evaluation of deep learning–based object detection models for marine debris detection using a downward-facing camera are central to this study. Multiple state-of-the-art detection architectures are implemented and compared to assess their suitability for real-time operation on embedded hardware under realistic aquatic visual degradation conditions.. The focus is placed on deployment feasibility and comparative performance rather than on the design of new detection architectures.
4. **Experimental Evaluation and Validation**
   The system is evaluated through a combination of controlled tests and limited real-world field experiments. Performance assessment emphasizes detection accuracy, inference latency on embedded hardware, and qualitative robustness under dynamic surface conditions. Field experiments are conducted to validate practical feasibility rather than to establish long-term operational statistics.

1.3.1 Exclusions

To maintain a focused and defensible scope, the following aspects are explicitly excluded from this thesis:

- Large-scale or long-duration field deployments and longitudinal environmental studies.
- Quantitative assessment of long-term ecological impact or debris removal effectiveness.
- Physics-based modeling of underwater or near-surface optical image formation.

- Development of novel object detection architectures or custom backbone networks.
- Swarm robotics, multi-vehicle coordination, or cooperative perception frameworks.
- Advanced hydrodynamic modeling or optimization of propulsion and hull dynamics.

These exclusions reflect deliberate design choices intended to prioritize **practical system integration and embedded perception performance** over algorithmic novelty or large-scale operational deployment. The defined scope ensures that the contributions of this thesis remain technically coherent, experimentally verifiable, and directly relevant to real-world ASV-based debris monitoring applications.

1.4 Structure of the Thesis

This thesis is structured into the following chapters:

- **Chapter 1 – Introduction**
  Provides an overview of the research background, motivation, objectives, and scope of the study, and outlines the overall structure of the thesis.
- **Chapter 2 – Literature Review**
  Reviews existing research on Autonomous Surface Vehicles (ASVs), marine navigation systems, and environmental monitoring applications, highlighting key challenges and research gaps.
- **Chapter 3 – Vision-Based Object Detection for Marine Environments**
  Examines object detection approaches relevant to aquatic environments, including classical methods, convolutional neural network–based detectors, transformer-based architectures, and publicly available marine debris datasets.
- **Chapter 4 – Design and Methodology**
  Describes the design and development of the ASV platform, including hardware selection, system integration, and the methodology adopted for onboard perception and navigation.
- **Chapter 5 – Implementation and Testing**
  Details the implementation of the ASV prototype, including fabrication, assembly, and staged testing procedures conducted in controlled and real-world environments.
- **Chapter 6 – Results and Discussion**
  Presents and analyzes the experimental results of the object detection system, comparing model performance and discussing deployment-related trade-offs and operational implications.
- **Chapter 7 – Conclusion and Future Work**
  Summarizes the research objectives, key findings, and contributions of the thesis, and outlines directions for future research.
- **Chapter 8 – References**
  Lists all cited works in accordance with the IEEE citation style.

Chapter 2: Literature Review

2.1 Overview of the Literature Review's Purpose

The purpose of this literature review is to establish the scientific and engineering context for developing an autonomous surface vehicle (ASV) equipped with a downward-looking camera for detecting marine debris in near-surface and shallow-water environments. The review consolidates prior work on (i) autonomous and unmanned water survey platforms, (ii) navigation technologies and their integration with perception, (iii) aquatic optical effects that degrade camera-based sensing, and (iv) modern deep-learning object detection architectures suitable for embedded, real-time deployment.

This chapter serves three objectives. First, it situates the proposed system within the evolution of marine survey vehicles and highlights enabling technologies that make low-cost autonomy feasible. Second, it synthesizes literature on vision-based debris detection, emphasizing operational constraints relevant to ASVs, including variable illumination, surface reflections, turbidity, and partial submergence. Third, it identifies the key gap motivating this thesis: object detectors trained on air-medium imagery often degrade significantly in aquatic scenes due to absorption and scattering effects, creating a domain mismatch that must be addressed through deployment-aware evaluation and careful architecture selection.

The literature review is structured to progressively narrow from general ASV platform capabilities to the specific technical challenges of deploying real-time object detection on embedded hardware under aquatic visual degradation. This organization directly informs the system design choices and experimental methodology presented in subsequent.

2.2 Existing Technologies in Water Survey Vehicles

2.2.1 Detailed Historical Perspective and Evolution

The development of water survey vehicles, particularly Autonomous Surface Vehicles (ASVs), has been a progressive journey that spans several decades. The initial impetus for the development of ASVs can be traced back to the late 20th century, driven primarily by military applications. The need for reconnaissance and surveillance in naval operations led to the early designs of unmanned surface vehicles that could navigate autonomously in various marine environments [4].

In the 1980s and 1990s, technological advancements in computing, sensors, and communication systems enabled significant improvements in ASV capabilities. The introduction of Global Positioning System (GPS) technology was a pivotal moment, allowing for precise navigation and positioning of these vehicles. Concurrently, the development of Inertial Measurement Units (IMUs) provided enhanced stability and orientation tracking, which were critical for autonomous operations [5].

The early 2000s saw a shift in the application of ASVs from exclusively military use to broader civilian and research applications. Environmental monitoring, hydrographic surveying, and search and rescue missions became key areas where ASVs demonstrated their utility. This shift was supported by advancements in sensor technology, such as multi-beam sonar systems and LiDAR, which allowed for detailed mapping and data collection[6].

In recent years, the integration of artificial intelligence (AI) and machine learning algorithms has further revolutionized ASVs. These technologies have enabled real-time data processing and decision-making, allowing ASVs to perform complex tasks such as dynamic obstacle avoidance and adaptive mission planning. The advent of edge computing has also facilitated onboard data processing, reducing the dependency on remote servers and improving the reliability and efficiency of ASV operations [7].

While early ASV development emphasized navigation and platform stability, recent progress has shifted toward perception-driven autonomy, where onboard sensing and machine learning enable vehicles to interpret complex marine environments in real time. This transition is particularly relevant for vision-based applications such as debris detection, where the ability to process camera data onboard—under variable lighting and water conditions—has become a defining capability of modern ASVs. Consequently, perception and embedded computation now represent core design drivers rather than auxiliary subsystems.

2.2.2 In-depth Comparison of Different Types of Survey Vehicles

Water survey vehicles can be broadly categorized into three main types: manned survey vessels, remotely operated vehicles (ROVs), and autonomous surface vehicles (ASVs). Each type has distinct characteristics, advantages, and limitations, making them suitable for different applications.

Manned Survey Vessels

Manned survey vessels are traditional boats or ships equipped with various instruments and operated by a crew. They have been the standard for hydrographic surveying and environmental monitoring for many decades. The primary advantage of manned vessels is the ability to carry a large array of equipment and personnel, enabling comprehensive data collection and real-time decision-making. However, they are costly to operate, limited by human endurance, and often unable to access shallow or hazardous areas safely [8].

Remotely Operated Vehicles (ROVs)

ROVs are unmanned submersibles controlled by operators from a surface vessel or shore. They are widely used for underwater inspections, repairs, and scientific research. ROVs offer several advantages, including the ability to operate at significant depths and the flexibility to carry various sensors and manipulators. However, they require a support vessel and a tether cable, which can limit their range and maneuverability [9].

Autonomous Surface Vehicles (ASVs)

ASVs are unmanned vessels that operate on the water's surface without direct human control. They are equipped with advanced navigation systems, sensors, and communication tools that enable autonomous operation. ASVs offer numerous advantages, including lower operational costs, the ability to operate in hazardous or inaccessible areas, and extended operational endurance. They are particularly useful for applications such as environmental monitoring, hydrographic surveys, and debris detection [6].

Compared to manned vessels and ROVs, ASVs provide a more flexible and efficient solution for many surveying tasks. Their autonomous capabilities allow for continuous operation without the need for a human crew, reducing costs and increasing the scope of possible missions. However, ASVs also face challenges such as the need for robust navigation systems and the ability to process data in real-time to adapt to changing environmental conditions.

From a perception standpoint, ASVs provide a unique balance between sensing capability and operational simplicity. Unlike ROVs, which require tethered operation and specialized launch platforms, ASVs can deploy vision sensors continuously with minimal logistical overhead. Compared to manned vessels, ASVs enable persistent, repeatable surveys at lower cost while avoiding safety risks in polluted or shallow environments. These characteristics make ASVs particularly suitable for scalable debris detection using downward-looking cameras, where frequent data collection and autonomous operation are essential.

2.2.3 Specific Case Studies and Examples with Analysis

Case Study 1: SeaCharger

The SeaCharger is an ASV designed for long-duration oceanographic missions. It is powered by solar panels and equipped with GPS for navigation. The SeaCharger has successfully completed several transoceanic voyages, collecting data on sea surface temperature, salinity, and marine life. This case demonstrates the potential of ASVs for extended missions, leveraging renewable energy sources to enhance sustainability and reduce operational costs [10].

Case Study 2: AutoNaut

AutoNaut is an ASV that uses wave propulsion for movement, making it highly energy-efficient. It is equipped with a range of sensors, including acoustic Doppler current profilers (ADCPs), meteorological sensors, and water quality sensors. AutoNaut has been used in various applications, including monitoring marine protected areas and conducting environmental impact assessments. This case highlights the versatility of ASVs and their ability to operate in diverse marine environments [11].

Case Study 3: ASV Global C-Worker 7

The C-Worker 7, developed by ASV Global, is a multi-purpose ASV used for hydrographic surveying, offshore asset inspection, and environmental monitoring. It is equipped with multi-beam echo sounders, side-scan sonars, and sub-bottom profilers. The C-Worker 7 has been deployed in various commercial and scientific missions, demonstrating the effectiveness of ASVs in complex and demanding operational scenarios [12].

Analysis

These case studies illustrate the wide range of applications and capabilities of ASVs. The SeaCharger exemplifies the use of renewable energy to achieve long-duration missions, while AutoNaut demonstrates innovative propulsion methods that enhance energy efficiency. The C-Worker 7 showcases the ability of ASVs to perform complex surveying tasks with high precision.

The analysis of these case studies underscores several key benefits of ASVs:

1.Operational Efficiency: ASVs can operate continuously without the need for a human crew, reducing costs and increasing mission duration.

2.Versatility: ASVs can be equipped with various sensors and instruments, making them suitable for a wide range of applications.

3.Accessibility: ASVs can access areas that are difficult or hazardous for manned vessels, such as shallow waters, ice-covered regions, and areas affected by pollution.

However, the successful deployment of ASVs also requires addressing several challenges, such as ensuring robust and reliable autonomous navigation, integrating advanced data processing capabilities, and developing sustainable energy solutions.

Although these platforms are not primarily designed for vision-based debris detection, they illustrate the operational envelope in which modern ASVs function: long endurance, minimal human intervention, and reliable autonomous navigation. These characteristics define the constraints under which onboard perception systems must operate. In particular, extended missions and energy efficiency favor lightweight, real-time perception pipelines, while diverse deployment environments highlight the need for robustness to environmental variability. These considerations directly inform the design requirements of ASV-mounted debris detection systems.

Conclusion

The evolution of water survey vehicles from manned vessels to advanced ASVs represents a significant technological advancement in marine research and environmental monitoring. ASVs offer numerous advantages, including lower operational costs, enhanced accessibility, and the ability to perform long-duration missions autonomously. By examining the historical development, comparing different types of survey vehicles, and analyzing specific case studies, this review highlights the transformative potential of ASVs in modern marine applications.

2.2.4 ASVs for Marine Debris Detection and Monitoring

Marine debris detection has emerged as a critical application domain for Autonomous Surface Vehicles (ASVs), driven by the growing environmental impact of plastic pollution in rivers, coastal regions, and harbors. Unlike traditional hydrographic surveying tasks, debris detection places significant emphasis on perception rather than mapping accuracy alone, requiring ASVs to reliably identify small, irregularly shaped objects under challenging optical conditions.

Early efforts in marine debris monitoring relied primarily on manual surveys and manned vessels, which are costly, labor-intensive, and limited in spatial and temporal coverage. The introduction of unmanned and autonomous platforms has enabled more frequent and scalable monitoring, particularly in areas that are hazardous, shallow, or economically impractical for crewed operations. ASVs are especially well suited for this role due to their ability to operate persistently at the water surface, where floating and near-surface debris is most prevalent.

Vision-Based Debris Detection on Surface Platforms

Vision-based sensing has become a dominant modality for debris detection on ASVs because of its low cost, rich semantic information, and compatibility with lightweight platforms. Cameras mounted in forward-looking or downward-looking configurations allow ASVs to observe floating debris as well as shallow submerged objects when water clarity permits. However, the effectiveness of camera-based detection is strongly influenced by environmental factors such as surface reflections, wave-induced motion, turbidity, and partial submergence of targets.

Several studies have demonstrated the feasibility of applying deep learning–based object detection models to marine debris imagery. The TrashCan dataset, introduced by Hong et al., represents one of the first large-scale, publicly available benchmarks for underwater debris detection, providing annotated images of trash objects captured in real marine environments. Baseline experiments using standard detectors (e.g., Faster R-CNN and Mask R-CNN) showed that while modern convolutional models can successfully detect debris, their performance degrades significantly compared to terrestrial benchmarks due to underwater optical distortions and class imbalance [13]. Although TrashCan focuses primarily on underwater imagery, its findings highlight challenges that also affect near-surface detection from ASVs, particularly reduced contrast and ambiguous object boundaries.

Large-Scale Robotic Debris Detection Initiatives

Beyond isolated academic studies, large-scale research initiatives have integrated perception-driven debris detection into autonomous marine systems. The SeaClear project, funded under the European Union's Horizon 2020 program, represents a comprehensive effort to develop autonomous robots capable of detecting, classifying, and collecting marine litter. SeaClear combines surface and underwater robotic platforms with vision-based perception pipelines to identify debris in real-world coastal environments. Project reports emphasize that reliable detection is one of the primary bottlenecks in autonomous litter removal, as visual conditions vary widely and debris objects often exhibit weak visual signatures [14]. The project's findings reinforce the need for robust perception algorithms that can operate under real deployment constraints rather than controlled laboratory conditions.

Similarly, the NOAA Marine Debris Program has investigated the use of unmanned and autonomous surface platforms for debris monitoring and assessment. Technical reports from NOAA highlight ASVs as promising tools for wide-area debris surveys, particularly in inland waterways and nearshore regions. These studies emphasize practical considerations such as sensor placement, onboard processing capability, and the trade-off between detection accuracy and real-time operation [15]. While many NOAA deployments focus on data collection rather than fully autonomous decision-making, they provide valuable insight into the operational realities faced by ASV-based debris monitoring systems.

Operational Constraints and Research Gaps

Despite demonstrated progress, several limitations remain in the current literature on ASV-based debris detection. First, many existing studies evaluate detection models on curated datasets or offline imagery, leaving uncertainty about real-time performance when models are deployed onboard resource-constrained platforms. Second, most detectors are pre-trained on large air-medium datasets (e.g., COCO), creating a domain gap that leads to reduced robustness when applied to aquatic imagery affected by absorption, scattering, and surface artifacts. Third, debris objects are often small, partially occluded, or visually similar to natural elements such as seaweed or foam, increasing false positive and false negative rates.

Another important gap concerns evaluation methodology. Many studies report standard accuracy metrics (e.g., mAP) without considering inference latency, memory footprint, or energy consumption, even though these factors directly affect the feasibility of long-duration ASV missions. As a result, there is a lack of systematic comparison between detector families—particularly lightweight one-stage models and emerging transformer-based approaches—under realistic ASV deployment constraints.

Relevance to This Thesis

The reviewed literature demonstrates that ASVs are a viable and increasingly important platform for marine debris detection, but also highlights unresolved challenges related to perception robustness and embedded deployment. These findings motivate the focus of this thesis on vision-based debris detection using a downward-looking camera mounted on an ASV, with particular emphasis on (i) mitigating the domain gap introduced by aquatic optical effects, and (ii) evaluating detection architectures in terms of both accuracy and real-time performance on embedded hardware. By addressing these gaps, this work aims to contribute practical insights toward reliable, scalable ASV-based debris monitoring systems.

2.3 Autonomous Navigation Systems and Integration of Navigation and Object Detection Systems

In the context of debris-detection ASVs, navigation and perception cannot be treated as independent subsystems. Object detection outputs directly influence local path planning, collision avoidance, and mission-level decisions, while navigation accuracy affects the spatial consistency and temporal stability of visual observations. Consequently, the integration of navigation and object detection must be designed as a closed-loop system in which sensing, state estimation, and control operate coherently under real-time and environmental constraints.

2.3.1 Comprehensive Overview of Navigation Technologies and System Integration

Autonomous navigation systems and the integration of navigation with object detection are foundational for the effective operation of Autonomous Surface Vehicles (ASVs). These

systems rely on a combination of advanced sensors, robust processing units, and sophisticated software algorithms to navigate and operate in complex marine environments.

Sensor Fusion

Sensor fusion integrates data from multiple sensors like GPS, IMUs, sonar, radar, and LiDAR to create a coherent understanding of the environment. Key techniques include:

Kalman filters: For probabilistic data fusion [16], [17].

Bayesian networks: For robust integration of sensor data [18], [17].

Recent Advances: Deep learning-based sensor fusion techniques are enhancing the robustness and accuracy of data integration [19].

For ASV-based debris detection, sensor fusion plays a critical role in stabilizing visual perception. GPS and IMU data provide motion estimates that can be used to compensate for platform movement, while fused state estimates improve temporal consistency when tracking detected debris across frames. This is particularly important for downward-looking cameras, where wave-induced motion and heading changes can cause rapid shifts in image perspective, leading to unstable detections if perception is treated in isolation.

Real-time Processing

ASVs require powerful processors, such as GPUs and FPGAs, for real-time data processing. Edge computing is crucial for local processing, reducing latency and dependency on remote servers. Frameworks like ROS help manage real-time processing efficiently [20]. Edge computing enhances responsiveness and processing speed, crucial for real-time operations [21].

Real-time processing constraints are especially stringent for ASVs operating in cluttered or debris-rich environments. Delayed perception can lead to late avoidance maneuvers or missed debris observations, particularly when the vehicle is operating at constant forward velocity. Edge computing enables perception and navigation decisions to be made onboard, reducing communication latency and increasing system reliability in environments where wireless connectivity is intermittent or unavailable. As a result, object detection models must be evaluated not only for accuracy but also for inference latency and computational footprint to ensure compatibility with real-time navigation loops.

Communication Protocols

Reliable communication is vital for seamless operation. Protocols like CAN bus, Ethernet, and wireless communications ensure smooth data flow between components [16]. Modern protocols like 5G are improving communication reliability and bandwidth for ASVs [22].

2.3.2 Detailed Discussion on Key Algorithms and Techniques

Simultaneous Localization and Mapping (SLAM)

SLAM algorithms are critical for ASVs, enabling them to build maps of unknown environments while simultaneously tracking their location. These algorithms use data from sensors like LiDAR, cameras, and IMUs to create and update maps. Key SLAM techniques include:

EKF-SLAM (Extended Kalman Filter SLAM): Uses probabilistic models for iterative position estimation and map updating [23].

FastSLAM: Combines particle filters with Kalman filters for efficient and accurate localization and mapping [18].

Graph-Based SLAM: Represents the environment as a graph, optimizing the map using graph-based techniques [16].

Recent Advances: ORB-SLAM3 includes improvements in visual-inertial navigation and loop closure [24].

While high-precision mapping is not always required for debris detection, SLAM provides spatial context that supports mission planning and repeatable survey patterns. In ASVs equipped with cameras, visual–inertial SLAM frameworks enable the association of detected debris with approximate spatial locations, facilitating mapping of debris distributions over time. Moreover, stable localization improves perception robustness by allowing temporal fusion of detections across frames, reducing false positives caused by transient visual artifacts.

Path Planning

Path planning algorithms determine the optimal route for an ASV from its current location to a desired destination. These algorithms consider environmental constraints, obstacle locations, and vehicle dynamics. Common techniques include:

A* Algorithm: A graph-based search algorithm that finds the shortest path between two points [20].

RRT (Rapidly-exploring Random Tree): A sampling-based algorithm useful in high-dimensional spaces [25].

Dijkstra's Algorithm: Explores all possible paths to find the shortest one, optimal but computationally intensive for large maps [16].

Recent Advances: Hybrid approaches combining A* with machine learning techniques for more efficient and adaptive path planning [26].

In perception-driven ASVs, path planning must remain reactive to object detection outputs, enabling dynamic re-planning when debris or obstacles are detected within the vehicle’s projected trajectory.

Obstacle Avoidance

Obstacle avoidance algorithms enable ASVs to detect and navigate around obstacles in real-time. These algorithms use sensor data to identify hazards and adjust the vehicle's path. Techniques include:

Potential Field Method: Treats obstacles as repulsive forces and the destination as an attractive force [18].

Dynamic Window Approach (DWA): Generates feasible trajectories considering the vehicle's dynamics and selects the safest path [25].

Artificial Neural Networks: Use machine learning to predict and avoid obstacles based on sensor data [20].

Recent Advances: Reinforcement learning allows ASVs to learn optimal avoidance strategies from simulation and real-world interactions [27].

Vision-based object detection provides complementary information to traditional proximity sensors by enabling semantic understanding of obstacles. For debris detection tasks, this allows the ASV to distinguish between navigational hazards and debris of interest, enabling different behavioral responses such as avoidance, tracking, or inspection. Integrating detection confidence into obstacle avoidance logic can further improve robustness by preventing spurious detections from triggering unnecessary maneuvers.

Challenges and Technological Solutions

Synchronization Issues

Synchronizing data from multiple sensors, which operate at different frequencies and latencies, is a significant challenge. Time-stamping data accurately and implementing synchronization algorithms, such as Precision Time Protocol (PTP), help align sensor data for coherent processing [16]. Advanced techniques using machine learning methods to predict and correct time discrepancies are emerging [28].

Environmental Variability

ASVs must operate in diverse environments with varying lighting, weather, and water conditions. Adaptive algorithms that adjust to changing conditions, such as dynamic thresholding for image processing or adaptive filtering for sonar data, are essential to maintain performance [20].

Computational Load

Simultaneous processing of navigation and object detection tasks demands substantial computational resources. Optimizing algorithms for efficiency and leveraging parallel processing capabilities of modern hardware can mitigate these challenges. Advances in edge computing also facilitate real-time processing, reducing reliance on cloud-based solutions and enhancing responsiveness [21].

2.3.3 Specific Examples of Implementations in ASVs

AutoNaut

AutoNaut is an ASV that uses wave propulsion for energy-efficient movement. It integrates GPS, IMUs, and SLAM algorithms for autonomous navigation, enabling long-duration missions for marine environmental monitoring without human intervention [29].

Wave Glider

Developed by Liquid Robotics, Wave Glider is designed for oceanographic data collection using wave and solar energy for propulsion. Its navigation system incorporates GPS, IMUs, and sonar, allowing it to navigate complex ocean environments efficiently [30].

C-Enduro

The C-Enduro, by L3 ASV, is a multi-purpose ASV for extended missions, equipped with LiDAR, radar, and cameras. It uses SLAM and advanced path planning for autonomous navigation, making it suitable for offshore wind farm inspections and marine wildlife monitoring [31].

Saildrone

Saildrone ASVs are used for climate data collection and have been deployed in extreme weather conditions to gather critical environmental data. Their advanced navigation and sensor systems enable them to operate autonomously in challenging environments [32].

These implementations demonstrate that modern ASVs combine navigation, sensing, and onboard processing into tightly integrated systems capable of long-duration autonomous operation. However, most existing platforms prioritize navigation and data collection, with perception often serving a secondary role. This highlights an opportunity for systems that explicitly elevate perception—particularly vision-based debris detection—to a first-class component of ASV autonomy.

Conclusion

Integrating navigation and object detection systems in ASVs involves addressing synchronization, environmental variability, and computational load. Advanced solutions in sensor fusion, real-time processing, and robust communication protocols ensure effective system performance. Real-world applications, from underwater exploration to marine debris detection and search and rescue operations, demonstrate the transformative impact of these integrated technologies. As technology advances, the potential for even greater innovation in ASVs grows, promising more efficient, reliable, and versatile applications ranging from environmental monitoring to complex maritime operations. Continued research and development in this field will further enhance the capabilities of ASVs, driving forward the future of autonomous marine technology.

2.4 Challenges in Underwater Optical Perception

Deploying optical sensors in aquatic environments introduces unique challenges not present in terrestrial computer vision. The performance of optical perception is fundamentally governed by the Inherent Optical Properties (IOPs) of the water column, specifically absorption and scattering [5].
For vision-based debris detection, these optical effects are not secondary disturbances but dominant factors that fundamentally alter image formation. Unlike terrestrial environments, where lighting variation is often the primary challenge, aquatic perception is governed by physical light–water

interactions that reduce contrast, distort color information, and introduce structured noise. As a result, object detection models trained on conventional air-medium datasets frequently experience severe performance degradation when deployed in underwater or near-surface aquatic environments, even when object classes are visually similar.

2.4.1 Absorption and Color Distortion

Water acts as a wavelength-selective filter. Long-wavelength light (red and orange) is attenuated rapidly, often disappearing within the first few meters of depth, while shorter wavelengths (blue and green) penetrate further. For neural networks, this results in a compressed color space where the discriminative power of RGB features is severely reduced, forcing models to rely more heavily on texture and shape—features that are themselves often degraded by turbidity [33].

From a learning perspective, wavelength-dependent attenuation compresses the effective color space available to convolutional neural networks. Features learned from RGB distributions in terrestrial datasets often rely on chromatic contrast that is absent or severely reduced underwater. Consequently, detectors are forced to rely on shape and texture cues that may themselves be degraded by turbidity and motion blur, increasing sensitivity to noise and background clutter.

2.4.2 Scattering and Marine Snow

Suspended particulate matter, including phytoplankton and microplastics, causes two distinct scattering phenomena. Forward scattering acts as a low-pass filter, blurring the high-frequency edge details necessary for object localization. Conversely, backscattering reflects ambient or artificial light back into the sensor, creating a "veiling light" effect often referred to as "marine snow," which significantly reduces contrast and signal-to-noise ratio [34]. These physical degradations create a substantial domain gap, causing models pre-trained on clear, air-medium datasets (like COCO) to fail when deployed underwater without domain-specific adaptation.

Forward scattering reduces high-frequency image content, weakening edges and fine structures that are critical for accurate bounding-box regression. Backscattering introduces an additive veiling component that lowers global and local contrast, producing haze-like artifacts commonly referred to as “marine snow.” For object detectors, these effects lead to unstable feature activations and inconsistent object boundaries, often resulting in fragmented detections or missed objects. The impact is particularly severe for small debris items, which may occupy only a few pixels and are easily lost amid scattering-induced noise.

2.4.3 Refraction, Surface Dynamics, and Camera Geometry

In ASVs equipped with downward-looking cameras, additional perceptual challenges arise from refraction at the air–water interface and dynamic surface conditions. Changes in wave slope alter the effective viewing angle, causing apparent object displacement and scale variation across frames. These distortions complicate temporal association of detections and can reduce the effectiveness of tracking-based filtering. Furthermore, surface reflections and sun glitter can saturate image regions, reducing the usable field of view and increasing false-negative rates. These effects are spatially and temporally non-uniform, making them difficult to remove through simple preprocessing and reinforcing the need for robust learning-based approaches.

2.4.4 Domain Gap Between Air-Medium and Aquatic Datasets

Most modern object detection architectures are pre-trained on large-scale datasets collected in air, such as COCO or ImageNet-derived benchmarks. These datasets lack the physical

degradations inherent to aquatic imaging, resulting in a domain gap when such models are deployed underwater or near the water surface. Empirical studies have shown that this mismatch leads to significant drops in detection accuracy and localization stability. Without adaptation, models tend to misclassify background clutter as debris or fail to detect partially submerged objects. This domain gap motivates the use of domain-specific datasets, domain-motivated augmentation (color/contrast/turbidity-like degradations)strategies, or fine-tuning procedures tailored to aquatic environments.

2.4.5 Implications for Vision-Based Debris Detection on ASVs

Collectively, absorption, scattering, refraction, and surface-induced distortions impose constraints that fundamentally shape the deployment of vision-based debris detection systems on ASVs. Effective perception pipelines must therefore operate under reduced color fidelity, weakened edges, structured noise, and dynamic appearance changes. These constraints indicate that standard accuracy metrics alone are insufficient to characterize real-world performance, as detection stability and responsiveness under visual degradation are equally critical. Accordingly, this thesis evaluates object detection models with an emphasis on robustness and real-time feasibility under aquatic imaging conditions, while acknowledging the practical limitations of training data and computational resources..

2.5 Machine Learning Models for Object Detection

2.5.1 Explanation of Various Models and Their Workings

Object detection, a fundamental task in computer vision, has undergone significant advancements thanks to machine learning, particularly deep learning. The journey began with traditional methods such as the Viola–Jones detector [35], which used Haar-like features and AdaBoost for face detection. These early techniques laid the groundwork for feature-based approaches, including the Histogram of Oriented Gradients (HOG) detector [36] and the Deformable Parts Model (DPM) [37].

The introduction of Convolutional Neural Networks (CNNs) revolutionized object detection. The R-CNN family, introduced by Girshick *et al.* [38], marked a major transition from handcrafted features to learned representations. R-CNN-based approaches generate region proposals and use CNNs to classify these regions. Faster R-CNN, proposed by Ren *et al.* [39], integrated the region proposal network directly into the CNN, significantly improving both speed and accuracy. Subsequent refinements, such as Fast R-CNN, further optimized the training and inference pipeline [40].

Single-shot detectors such as YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector) further advanced real-time object detection. YOLO, introduced by Redmon *et al.* [41], processes images in a single evaluation, predicting bounding boxes and class probabilities simultaneously. This design enables high inference speed and makes YOLO-based detectors well suited for real-time robotic and embedded applications. SSD, proposed by Liu *et al.* [42], employs multiple feature maps at different resolutions, improving detection performance for objects of varying scales while maintaining computational efficiency.

Recent advancements include transformer-based models such as DETR (Detection Transformer), introduced by Carion *et al.* [43]. DETR reformulates object detection as a direct set-prediction problem using attention mechanisms, simplifying the detection pipeline by eliminating handcrafted components such as anchor boxes and non-maximum suppression. Extensions such as Rank-DETR further refine prediction ranking to improve localization accuracy, demonstrating the growing relevance of transformer-based approaches in object detection [44].

Transfer Learning

Transfer learning has become an essential technique in deep learning, particularly for object detection tasks. It involves initializing models with weights pre-trained on large-scale datasets such as ImageNet and subsequently fine-tuning them on task-specific datasets. This approach reduces training time and improves performance when labeled data are limited.
Pre-trained backbone models such as AlexNet, VGG, ResNet, and Inception are commonly used in object detection architectures. AlexNet, introduced by Krizhevsky *et al.* [45], demonstrated the effectiveness of deep CNNs for large-scale visual recognition. VGG networks, proposed by Simonyan and Zisserman [46], emphasized architectural simplicity and depth through the use of small convolutional filters. ResNet, introduced by He *et al.* [47], employed residual connections to enable the training of very deep networks and mitigate the vanishing gradient problem. The Inception architecture, proposed by Szegedy *et al.* [48], incorporated multi-scale convolutions within a single network to improve representational efficiency.
In this thesis, transfer learning is employed primarily as a practical initialization strategy rather than as a subject of investigation in its own right. Pre-trained weights provide a stable starting point for training object detectors on limited aquatic datasets, reducing convergence time and improving baseline performance. No attempt is made to modify backbone architectures or optimize feature extractors specifically for underwater imaging; instead, the focus is placed on evaluating detector-level performance under realistic deployment constraints. While the architectural evolution of object detection models has been driven largely by benchmarks in terrestrial computer vision, many of the underlying design principles—such as feature reuse, multi-scale representation, and end-to-end optimization—remain directly relevant to autonomous marine platforms. However, real-time inference requirements, limited onboard computation, and visually degraded inputs constrain the direct applicability of many high-capacity models to ASV-based debris detection.

2.5.2 Comparison of Different Architectures and Their Effectiveness

Two-

Object detection architectures are commonly classified into two-stage and one-stage detectors; however, more recent approaches introduce transformer-based, set-prediction frameworks that do not conform to this traditional taxonomy. Two-stage methods (e.g., Faster R-CNN) generate region proposals before classification, generally improving localization accuracy at higher computational cost. One-stage methods (e.g., YOLO, SSD, RetinaNet) directly predict bounding boxes and class probabilities in a single pass, enabling real-time inference on edge devices. Transformer-based detectors (e.g., DETR and real-time variants) reformulate detection as direct set prediction using attention mechanisms, reducing reliance on hand-designed components such as anchors and non-maximum suppression, at the cost of higher memory demand and deployment complexity on constrained hardware.

Two-stage Detectors

R-CNN Family (R-CNN, Fast R-CNN, Faster R-CNN):

These models first generate region proposals and then classify them. They offer high accuracy but are computationally intensive. Mask R-CNN extends Faster R-CNN by adding a branch for predicting object masks, allowing for precise instance segmentation.

Faster R-CNN:

Improved upon its predecessors by integrating the region proposal network with the detection network, enhancing both speed and accuracy.

Mask R-CNN:

Adds an additional mask prediction branch to Faster R-CNN, enabling pixel-level segmentation of detected objects.

One-stage Detectors

YOLO-based detectors are widely adopted in real-time robotic systems due to their single-stage formulation and favorable speed–accuracy trade-offs. Successive YOLO variants introduce architectural refinements intended to improve efficiency and detection robustness; however, reported gains are typically benchmark-dependent and may not directly translate to aquatic environments. For ASV deployment, lightweight YOLO configurations are of particular interest, as they enable real-time inference on embedded hardware. Consequently, model selection in this thesis is based on empirical performance evaluation under aquatic visual degradation rather than on architectural claims alone [50].

SSD (Single Shot MultiBox Detector):

Uses multiple feature maps for detection, enhancing accuracy for small objects while maintaining high speed.

RetinaNet:

Introduced the Focal Loss to address class imbalance, improving performance for detecting rare objects [49].

Transformer-based Models

While Convolutional Neural Networks (CNNs) have traditionally dominated real-time detection, 2025 has seen the maturation of Real-Time Detection Transformers (RT-DETR). Unlike CNNs, which use local receptive fields, Transformers utilize self-attention mechanisms to model global scene dynamics [43], [51]. These architectures leverage a hybrid encoder to process multi-scale features, allowing them to effectively separate signal from noise in turbid water. However, despite their accuracy advantages, Transformers generally impose a higher memory footprint than CNNs, presenting a trade-off between semantic understanding and raw inference speed on edge devices like the Raspberry Pi 5 [52]. For this reason, transformer-based detectors are considered primarily as comparative baselines rather than as default deployment candidates in the proposed ASV system.

2.5.3 Case Studies and Real-World Applications

Autonomous Robotic Platforms

Autonomous robotic platforms rely on object detection as a core perception capability to enable real-time navigation, obstacle avoidance, and interaction with dynamic environments. In mobile robots and unmanned systems, object detection outputs are typically fused with localization and tracking modules to support closed-loop control and safe motion planning. The choice of detection architecture in these systems is strongly influenced by constraints on onboard computation, power consumption, and inference latency.

Two-stage detectors, such as Faster R-CNN, are widely recognized for their strong localization accuracy and robustness in complex scenes; however, their multi-step processing pipeline and higher computational demands often limit their deployment in real-time, embedded scenarios [53]. Conversely, one-stage detectors, including YOLO, perform detection in a single forward pass, significantly reducing latency and making them suitable for platforms requiring rapid perception updates [54]. This trade-off between accuracy and efficiency is central to autonomous navigation and is directly applicable to ASVs operating in constrained and dynamic aquatic environments.

Maritime and Aquatic Robotics Applications

In maritime and aquatic robotics, object detection supports critical tasks such as obstacle avoidance, situational awareness, and environmental monitoring. ASVs operating in coastal regions, rivers, and harbors encounter a diverse set of hazards, including floating debris, partially submerged objects, and man-made structures. When using downward-facing or oblique cameras, these systems must detect targets whose visual appearance is altered by refraction at the air–water interface, wave-induced motion, surface glare, and variable illumination [55].

Several studies in marine robotics highlight that visual perception in aquatic environments is inherently more challenging than in terrestrial settings. Water-induced distortions reduce contrast and color fidelity, while small debris objects may appear fragmented or intermittently visible due to surface dynamics [56]. As a result, perception systems must balance robustness and responsiveness, favoring detectors that can operate reliably under degraded visual conditions while maintaining real-time performance. Empirical evidence suggests that lightweight detection models are more suitable for continuous onboard operation on ASVs, whereas higher-capacity models are often reserved for offline analysis or post-mission data processing [57].

Agriculture and Environmental Monitoring

In precision agriculture and environmental monitoring, object detection is widely used on unmanned aerial vehicles (UAVs) and ground robots to identify crops, weeds, pests, and signs of vegetation stress. These platforms operate under constraints similar to ASVs, including limited computational resources, reliance on onboard power, and exposure to variable environmental conditions such as changing illumination, shadows, and background clutter.

Studies in this domain demonstrate that one-stage detectors are frequently preferred due to their favorable balance between detection accuracy and inference speed [58]. Additionally, the success of these models in visually complex outdoor environments underscores their robustness to non-ideal imaging conditions. This is particularly relevant for ASV-based debris detection, where reflections, water turbidity, and surface texture introduce noise patterns analogous to those encountered in agricultural scenes.

Wildlife and Ecological Monitoring

Object detection has also been extensively applied in wildlife and ecological monitoring, especially through the use of camera traps and autonomous sensing platforms. In these applications, detectors must identify animals across wide variations in pose, scale, partial occlusion, and background complexity. Models such as SSD and YOLO variants have been successfully deployed to automate species detection and population monitoring, significantly reducing the need for manual data annotation [59].

The relevance of these studies to aquatic debris detection lies in their emphasis on robustness to domain-specific visual degradation. Similar to wildlife monitoring, debris detection in aquatic environments often involves small targets with irregular shapes that may blend into complex backgrounds. Lessons from ecological monitoring therefore inform the design of perception pipelines capable of handling sparse, noisy, and highly variable visual cues.

Cross-Domain Insights for ASV-Based Debris Detection

These cross-domain case studies collectively motivate the need for systematic evaluation of object detection architectures under realistic deployment constraints. In particular, they support the investigation of lightweight one-stage detectors and emerging transformer-based models within a deployment-aware evaluation framework that considers aquatic visual degradation. This perspective directly informs the experimental methodology adopted in this thesis, which evaluates detection performance not only in terms of accuracy but also latency and deployability on embedded ASV hardware.

Across these application domains, a consistent pattern emerges: object detection systems deployed on autonomous platforms must balance detection accuracy with latency, energy consumption, and robustness to environmental variability. Although the operational contexts differ, the constraints faced in autonomous driving, agriculture, and ecological monitoring closely parallel those encountered by ASVs. These cross-domain experiences reinforce the relevance of lightweight, real-time detection architectures for debris detection tasks and motivate the experimental focus adopted in this thesis.

2.6 Object Detection in Autonomous Robotic Platforms: Cross-Domain Applications and Architectural Trade-offs

Object detection is a foundational capability in autonomous robotic platforms, directly influencing real-time navigation, obstacle avoidance, and interaction with dynamic environments. For Autonomous Surface Vehicles (ASVs), the ability to accurately and efficiently detect and localize objects—ranging from floating debris and marine fauna to other vessels—is critical for safe and adaptive operation in complex aquatic environments. The same principles extend to terrestrial and aerial robotics, where object detection enables unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) to traverse cluttered landscapes, avoid obstacles, and interact with both static and moving elements in their surroundings.

5.6.1 Detection Architecture Trade-offs: Two-Stage vs One-Stage Detectors

The selection of object detection architecture is governed by the trade-off between accuracy, inference speed, and computational efficiency. Two-stage detectors such as Faster R-CNN first generate region proposals and then classify them, achieving high accuracy particularly in complex scenes but often at the expense of latency and computational overhead. This makes them less suited for real-time deployments on resource-constrained embedded platforms typically found in ASVs and edge robotics. In contrast, one-stage detectors like YOLO and SSD eliminate the proposal stage, directly predicting bounding boxes and classes in a single pass, thus offering superior inference speed and lower memory requirements. While early one-stage models sacrificed some accuracy for speed, recent advances—such as YOLOv4, YOLOv5, and transformer-based architectures—have narrowed this gap, enabling robust real-time detection even under challenging visual conditions.[60], [61]

5.6.2 Applications in Maritime and Aquatic Robotics

In maritime robotics, object detection must contend with unique environmental challenges such as variable lighting, color attenuation, scattering, and turbidity. These factors degrade visual inputs, necessitating models that are both resilient to noise and efficient enough for on-board processing. Lightweight YOLO configurations have demonstrated favorable real-time performance on platforms like the Raspberry Pi, balancing the need for speed and accuracy in aquatic debris detection and collision avoidance. Transformer-based models, leveraging self-attention mechanisms, offer improved semantic understanding but typically require greater memory and computational resources, introducing trade-offs in edge deployment.[60], [62], [63]

5.6.3 Object Detection in Agriculture and Environmental Monitoring

Object detection technologies developed for ASVs share parallels with those used in agricultural robotics and environmental monitoring. UAVs and UGVs employ similar detection models to identify crops, weeds, livestock, and environmental hazards, often operating in outdoor settings subject to illumination changes, occlusions, and clutter. The lessons learned in optimizing models for speed and robustness in these domains directly inform the selection and tuning of detectors for aquatic platforms, where rapid response and low-latency processing are equally critical.

5.6.4 Wildlife and Ecological Monitoring

In wildlife and ecological monitoring, object detection enables automated tracking of animal populations, behavioral studies, and identification of invasive species. ASVs equipped with advanced detection architectures can contribute to these efforts by surveying aquatic habitats, monitoring species distributions, and detecting debris or pollutants that threaten ecosystem health. The cross-domain applicability of detection models underscores the importance of selecting architectures that balance accuracy, efficiency, and resilience to environmental variability.[60]

5.6.5 Cross-Domain Insights and Implications for ASV-Based Debris Detection

Synthesis of cross-domain insights reveals that optimizing object detection for ASV deployment involves more than architectural claims; it requires empirical evaluation under representative conditions, including underwater and near-surface visual degradations. Lightweight models such as YOLO are often preferred for embedded systems due to their favorable speed–accuracy trade-offs, while the adoption of transformer-based models is motivated by their ability to model global scene context in noisy environments. The balance between semantic understanding, inference speed, and memory footprint ultimately guides the experimental methodology for evaluating detection architectures in ASV-based debris detection.[60], [61], [62], [63]

5.6.6 Implications for Experimental Methodology

The diverse requirements and constraints of maritime, agricultural, and ecological environments motivate a holistic evaluation approach. This thesis therefore focuses on lightweight and transformer-based models, leveraging domain-specific augmentation pipelines and empirical testing under aquatic conditions to inform the selection of optimal architectures for ASV-based debris detection. Integrating cross-domain experiences ensures that the chosen detection solutions are robust, efficient, and adaptable to the operational realities of autonomous surface vehicles.[61], [62]

Beyond architectural and deployment considerations, the broader relevance of ASV-based perception systems is reflected in their potential contribution to sustainability-oriented monitoring and environmental assessment, which is discussed in the following section.

2.7 Environmental Impact and Sustainability

2.7.1 Role of Autonomous Surface Vehicles in Environmental Sustainability

Autonomous Surface Vehicles (ASVs) contribute to environmental sustainability primarily by enabling persistent, data-driven monitoring of aquatic environments with reduced operational cost and energy consumption compared to conventional crewed vessels [68]. Their autonomous operation allows repeated surveys of coastal waters, rivers, and lakes, providing high temporal and spatial resolution data that support long-term environmental assessment and management.

From an engineering perspective, the sustainability relevance of ASVs is not derived from direct remediation or cleanup alone, but from their ability to support informed decision-making through continuous sensing and perception. By integrating onboard navigation, sensing, and perception systems, ASVs reduce the need for fuel-intensive manned missions while minimizing human exposure to hazardous or polluted environments [64]. This operational efficiency directly aligns with sustainability objectives by lowering emissions and resource usage associated with environmental monitoring campaigns.

Vision-based perception systems further enhance this contribution by enabling automated identification of surface-level phenomena such as floating debris, oil slicks, and biological matter. Rather than relying on manual inspection or post-mission analysis, onboard object detection allows ASVs to collect targeted environmental data in real time, improving situational awareness and enabling adaptive survey strategies.

2.7.2 Sustainability-Oriented ASV Applications

Several large-scale research and operational programs demonstrate the role of ASVs as enabling platforms for sustainability-oriented monitoring. The NOAA Marine Debris Program has explored the use of autonomous and semi-autonomous surface vehicles for surveying inland waterways and coastal regions, emphasizing detection, mapping, and characterization of marine debris rather than direct removal [64]. These efforts highlight the importance of reliable perception systems capable of operating under variable lighting, water conditions, and cluttered backgrounds.

Similarly, initiatives such as The Ocean Cleanup have investigated autonomous platforms for reconnaissance and monitoring tasks, including identifying debris accumulation zones and evaluating cleanup system performance [65]. While primary debris removal systems operate at scale, autonomous vehicles provide complementary sensing capabilities that improve operational efficiency and situational awareness.

ASVs are also widely used for water quality monitoring in freshwater systems, including lakes and reservoirs, where repeated autonomous surveys provide data on pollutants, algal blooms, and invasive species. In these applications, sustainability benefits arise from long-duration operation and reduced reliance on crewed sampling campaigns. Optical and multisensory

perception systems play a key role by enabling automated detection of surface anomalies and environmental indicators.

In marine habitat monitoring, ASVs support non-invasive observation of sensitive ecosystems such as coral reefs and coastal habitats. By collecting visual and sensor data without physical disturbance, ASVs contribute to environmental assessment while minimizing ecological impact [67]. These applications reinforce the view that sustainability-oriented ASV deployments depend strongly on robust perception rather than intervention-based mechanisms.

2.7.3 Engineering Considerations for Sustainable ASV Deployment

Sustainability considerations influence ASV system design at multiple levels, including propulsion efficiency, sensing strategy, and computational workload. Lightweight hull designs, low-power propulsion systems, and efficient mission planning reduce energy consumption and extend operational duration. Embedded processing enables onboard analysis of sensor data, reducing the need for continuous high-bandwidth communication and associated energy costs.

From a perception standpoint, the choice of object detection architecture has direct sustainability implications. Computationally intensive models increase power consumption and limit mission duration, whereas lightweight detectors enable real-time operation on embedded hardware with lower energy demand. Consequently, perception systems must balance detection accuracy with computational efficiency to support sustained autonomous operation.

It is important to note that advanced sustainability concepts such as fully autonomous debris removal, renewable-energy-powered ASVs, and cooperative swarm systems remain active research topics. While these approaches show long-term potential, they are largely beyond the scope of the system developed in this thesis. Instead, the focus is placed on perception-driven monitoring as a foundational capability that supports sustainable environmental management strategies.

2.7.4 Implications for This Thesis

Within the scope of this work, environmental sustainability is addressed indirectly through the development and evaluation of a vision-based object detection pipeline suitable for embedded ASV deployment. By prioritizing real-time feasibility, robustness under aquatic visual degradation, and low computational overhead, the proposed approach aligns with sustainability-oriented objectives such as extended mission duration and reduced operational cost.

The literature reviewed in this section supports the conclusion that reliable perception is a critical enabling factor for environmentally sustainable ASV operations. While object detection alone does not resolve marine pollution challenges, it provides essential data that support targeted intervention, policy development, and long-term environmental assessment. This perspective informs the experimental methodology adopted in this thesis, where detection models are evaluated not only in terms of accuracy, but also with respect to their practicality for sustained autonomous operation in real-world aquatic environments [66].

2.8 Future Trends and Research Directions

2.8.1 Trends Relevant to Vision-Based ASV Perception

Recent research in autonomous marine systems increasingly emphasizes practical deployability and system-level robustness, rather than purely algorithmic novelty or benchmark-driven performance gains [69], [70]. This shift is particularly evident in perception systems, where the constraints imposed by embedded computation, limited power budgets, and challenging aquatic visual conditions significantly affect real-world applicability.

In the domain of object detection, lightweight one-stage detectors, especially YOLO-based architectures, remain widely adopted in autonomous robotic platforms due to their favorable balance between inference speed and detection accuracy [69], [71]. These models are commonly deployed in scenarios where real-time perception is required under strict latency and energy constraints, making them suitable candidates for ASV-based debris detection.

In parallel, transformer-based object detectors, such as DETR and its real-time variants, have gained attention for their ability to model global scene context through self-attention mechanisms [72], [73], [74]. While these models often demonstrate strong performance in terms of localization and semantic understanding, their higher computational and memory requirements currently limit their suitability for continuous deployment on small, resource-constrained ASVs.

Another important trend highlighted in the literature is the recognition that performance on terrestrial benchmark datasets does not reliably translate to aquatic environments. Factors such as surface glare, refraction, turbidity, and dynamic backgrounds introduce domain shifts that significantly degrade detection performance if not explicitly considered during evaluation [75], [76]. As a result, recent studies increasingly advocate for deployment-oriented evaluation protocols that incorporate realistic environmental conditions and system constraints.

2.8.2 Implications for Object Detection Model Selection

The trends identified in the literature suggest that future progress in ASV perception will depend less on the introduction of novel detection architectures and more on systematic evaluation of

existing models under realistic deployment constraints [69], [70]. For ASV-based debris detection, this includes assessing inference latency on embedded hardware, robustness to aquatic visual degradation, and stability of detections over time.

In line with these considerations, this thesis adopts a comparative evaluation approach, focusing on lightweight CNN-based detectors and representative transformer-based models rather than proposing new architectures. YOLO-based models are selected due to their established use in real-time robotic perception, while transformer-based detectors such as DETR are included as comparative baselines to assess potential gains in semantic modeling and global context understanding [69], [71], [72].

The use of a downward-facing camera configuration, representative of practical ASV deployments, further aligns the experimental design with real-world operational conditions. This choice reflects the growing emphasis in the literature on evaluating perception systems within realistic sensing geometries and mission scenarios, rather than idealized experimental setups.

2.8.3 Open Challenges Identified in the Literature

Despite advances in object detection, robust generalization under aquatic visual degradation remains a significant open challenge. Variations in illumination, surface motion, and water optical properties continue to limit the reliability of vision-based perception systems, particularly when models are trained primarily on terrestrial imagery [75], [76].

Another unresolved issue concerns the energy–accuracy trade-off inherent in deploying deep learning models on embedded ASV platforms. While higher-capacity models may achieve improved detection accuracy, their computational demands can substantially reduce mission duration on battery-powered systems, undermining long-term monitoring objectives [77], [74].

Additionally, the literature highlights a lack of standardized evaluation protocols for ASV perception systems. Differences in camera placement, sensing geometry, environmental conditions, and performance metrics complicate direct comparison across studies and hinder reproducibility [75], [70]. Addressing this gap requires evaluation methodologies that explicitly consider deployment constraints and operational realism.

2.8.4 Positioning of This Thesis

Within the broader research landscape, this thesis addresses a subset of the identified challenges by focusing on deployment-aware evaluation of object detection models for ASV-based debris

detection. Rather than developing new perception architectures or training paradigms, the contribution lies in empirically assessing existing models under realistic aquatic conditions and embedded hardware constraints [69], [70].

By emphasizing real-time feasibility, robustness to visual degradation, and computational efficiency, this work aligns with emerging research directions that prioritize operational relevance over benchmark-centric optimization. The results presented in the following chapter provide practical insights into the suitability of different object detection architectures for sustained autonomous operation, thereby establishing a clear link between the literature review and the experimental methodology adopted in this thesis.

Based on the constraints identified in the literature, the experimental evaluation in this thesis focuses on a comparative assessment of lightweight one-stage detectors from the YOLO family and a representative transformer-based detector (RT-DETR). Model performance is evaluated using standard detection metrics, including precision, recall, mAP@50, and mAP@50–95, alongside inference latency and estimated throughput, reflecting the trade-offs between accuracy and real-time feasibility emphasized in prior ASV perception studies. Latency measurements are initially obtained on a high-performance GPU to establish architectural efficiency, while embedded deployment constraints motivate additional analysis on resource-limited platforms such as the Raspberry Pi 5. These metrics are selected directly in response to the documented domain gap between terrestrial and aquatic imagery and the computational limitations of onboard ASV systems, where detection robustness and responsiveness are equally critical for practical operation.

Chapter 3: Machine Learning for Object Detection

3.1 Overview of Object Detection

Object detection represents a fundamental pillar of computer vision, encompassing the dual task of classification—identifying the semantic category of an entity—and localization—determining its spatial position within an image, typically represented by axis-aligned bounding boxes. In the context of autonomous maritime robotics, object detection functions as the perceptual layer that converts raw visual data into semantically meaningful information suitable for downstream decision-making.

Early object detection systems relied on handcrafted feature descriptors, such as Scale-Invariant Feature Transform (SIFT) and Histograms of Oriented Gradients (HOG), combined with classical classifiers including Support Vector Machines (SVMs). While effective in constrained terrestrial scenarios, these approaches exhibit limited robustness when exposed to the non-linear noise, light attenuation, and spectral distortions characteristic of aquatic environments, as discussed in Chapter 5.

The emergence of deep learning, particularly Convolutional Neural Networks (CNNs) and more recently transformer-based architectures, has fundamentally transformed object detection by enabling end-to-end feature learning. These models learn hierarchical representations directly from data, allowing them to adapt to complex visual degradations such as turbidity, surface glare, marine snow, and chromatic aberration. Vision Transformers further extend this capability by modeling long-range spatial dependencies, improving contextual reasoning in cluttered scenes.

Within this thesis, the object detection pipeline is designed to satisfy three deployment-driven requirements:

1. **High inference throughput**, enabling real-time operation on embedded hardware such as the Raspberry Pi 5.
2. **Environmental robustness**, ensuring reliable discrimination between anthropogenic debris and biological entities under visually degraded aquatic conditions.
3. **Edge optimization**, balancing detection accuracy (mAP) against computational complexity, power consumption, and thermal constraints imposed by an Autonomous Surface Vehicle (ASV).

This chapter presents the software frameworks and model architectures used to meet these requirements. Particular emphasis is placed on the YOLO (You Only Look Once) detection paradigm and the RT-DETR (Real-Time Detection Transformer) architecture, culminating in the

selection of YOLO11n as the primary detection model based on its favorable accuracy–latency trade-off under embedded deployment constraints.

3.2 Implementation Tools and Frameworks

The realization of a real-time perception system for an Autonomous Surface Vehicle requires a software ecosystem capable of supporting both experimental flexibility and deployment reliability. The tools selected in this work span multiple abstraction layers, from deep learning research frameworks to real-time image processing libraries and embedded inference engines. Selection criteria included computational efficiency, cross-platform compatibility, and predictable runtime behavior under resource constraints.

3.2.1 Deep Learning Ecosystems: TensorFlow and Keras

TensorFlow and its high-level API, Keras, have historically served as foundational platforms for large-scale machine learning development and deployment. TensorFlow provides an extensive ecosystem encompassing model training, serving, and optimization, including TensorFlow Lite for mobile and embedded inference [78], [79].

During the preliminary phase of this research, Keras was employed for rapid architectural prototyping, leveraging its modular design to explore lightweight backbone networks such as MobileNet variants. However, despite the maturity of the TensorFlow ecosystem, its relatively rigid execution model and higher integration overhead for custom layers posed limitations for rapid experimentation in a research-driven marine robotics context.

3.2.2 PyTorch and the Ultralytics Research Framework

For the final implementation of the object detection pipeline, PyTorch was selected as the primary deep learning framework [80]. PyTorch's imperative execution model and dynamic computation graph enable intuitive debugging and flexible architectural modification, which are particularly advantageous during iterative model evaluation and deployment tuning.

This thesis utilizes the Ultralytics framework, which builds upon PyTorch to provide a specialized research and training environment for the YOLO family of detectors. Ultralytics abstracts complex training procedures—such as optimizer configuration, learning-rate scheduling, and early stopping—allowing the research focus to remain on domain-specific challenges associated with aquatic imagery. Recent compiler-level optimizations introduced via torch.compile() further reduce execution overhead, narrowing the performance gap between dynamic and static graph frameworks during both training and inference.

3.2.3 Real-Time Data Handling via OpenCV

While neural networks provide semantic interpretation, real-time image acquisition and visualization are handled through the Open Source Computer Vision Library (OpenCV) [81]. OpenCV remains a de facto standard for managing video pipelines in autonomous systems due to its performance, portability, and extensive hardware acceleration support.

In this work, OpenCV is employed for asynchronous frame acquisition from the onboard camera, spatial preprocessing operations such as resizing and color-space conversion, and real-time visualization of detection outputs. Bounding box overlays and confidence annotations are rendered with minimal overhead, enabling operator monitoring without interfering with the onboard inference pipeline.

3.2.4 Inference Acceleration and Deployment Formats

To satisfy real-time performance requirements on the Raspberry Pi 5, multiple deployment-oriented inference formats were considered. Native PyTorch execution provides a flexible baseline but incurs non-negligible memory and computational overhead on ARM-based platforms.

As part of the deployment exploration, models were exported to the Open Neural Network Exchange (ONNX) format, which serves as a framework-agnostic representation for inference optimization and portability [82]. For ARM-based embedded platforms, the NCNN inference framework was identified as a potential optimization path due to its hand-tuned support for ARM NEON vector instructions and reduced inference latency variability [83].

Additionally, the OpenVINO toolkit was considered as an alternative optimization pathway for potential future deployment on Intel-based embedded systems, where hardware-specific optimizations such as reduced-precision inference may offer performance benefits [84]. These acceleration frameworks are discussed as deployment options; however, quantitative performance evaluation in this thesis focuses on native PyTorch-based inference to maintain experimental consistency across platforms.

3.3 6.3 Strategic Dataset Curation

The effectiveness of vision-based object detection systems for Autonomous Surface Vehicles (ASVs) is strongly influenced by the characteristics of the training data. Unlike terrestrial perception tasks, aquatic environments introduce unique challenges related to camera geometry, lighting variability, surface reflections, and dynamic backgrounds. As a result, dataset selection for ASV deployment cannot be treated as a purely data-driven exercise, but must instead be guided by system-level considerations that reflect real operational constraints.

In the context of this thesis, dataset curation is approached as an engineering design decision rather than a benchmark-driven optimization task. Factors such as sensor viewpoint compatibility, annotation consistency, target class relevance, and dataset scale are prioritized to

ensure alignment with the intended deployment scenario. Particular emphasis is placed on the suitability of datasets for embedded, real-time inference, where computational efficiency and robustness under visual degradation are critical.

Given the diversity of available marine and underwater vision datasets, a comparative analysis is first conducted to assess their applicability to ASV-based surface debris detection. This analysis informs a disciplined dataset selection process that balances experimental feasibility with representativeness of real-world operating conditions. The outcome of this process directly shapes the subsequent preprocessing, training, and evaluation stages of the perception pipeline developed in this work.

3.3.1 Comparative Analysis of Candidate Datasets

Several datasets have been proposed for underwater and marine-litter perception tasks; however, they differ substantially in sensing geometry, task scope, annotation structure, and environmental conditions. For ASV-based debris detection, these differences directly affect the validity of model training and the interpretability of deployment results. Accordingly, this thesis reviews representative candidate datasets—SeaClear, Trash-ICRA19, Brackish, and TrashCan—to determine their suitability for a surface-vehicle perception pipeline.

The SeaClear initiative (Horizon 2020) targets end-to-end marine litter detection and collection using heterogeneous robotic agents and field deployments, and includes data products aligned with project-level objectives and system integration requirements. While valuable as a robotics benchmark, its dataset assets are designed to support a broader intervention pipeline rather than a narrowly scoped, vision-only detector trained and evaluated under embedded ASV constraints.

The Trash-ICRA19 dataset provides bounding-box annotations for underwater trash and has been used as an early benchmark for underwater debris detection. However, its imaging conditions and viewpoint assumptions are more closely aligned with submerged platforms (e.g., ROV/AUV use cases) than with near-surface ASV operation. In addition, its scale and class taxonomy are comparatively limited relative to more recent marine-debris datasets, which can constrain training stability and generalization when evaluating modern detectors.

The Brackish dataset was designed for perception in turbid brackish-water environments and is annotated primarily for biological targets (e.g., fish, crabs, and other marine organisms), making it relevant for ecological monitoring and tracking under severe visual degradation. However, it does not provide a debris-focused class taxonomy suitable for training debris detectors, and its target distribution and labeling intent do not match the debris-detection objective of this thesis. For this reason, Brackish is treated as a reviewed-but-excluded candidate dataset rather than a training or evaluation corpus.

The TrashCan dataset is a large-scale marine debris dataset designed for underwater trash detection and segmentation, and it is widely referenced in recent marine perception literature. Importantly for this thesis, TrashCan supports debris-oriented learning with richer annotations and broader appearance diversity than earlier benchmarks such as Trash-ICRA19. In the configuration used in this work, TrashCan comprises 7,212 labeled images, which provides sufficient scale for training lightweight detectors while maintaining manageable preprocessing and training complexity for deployment-aware experimentation.

Taken together, this comparative review indicates that not all “marine” datasets are equally suitable for embedded ASV perception studies. Datasets optimized for intervention pipelines (e.g., SeaClear) or

submerged viewpoints (e.g., Trash-ICRA19) may be valuable for other research objectives, but they introduce mismatches in task definition and sensing conditions when the goal is deployment-aware evaluation of real-time debris detection on an ASV. Conversely, TrashCan provides the closest alignment to a debris-centric detection task with sufficient scale and community adoption to support meaningful comparisons.

Table 3.1: Dataset comparative review table.

Dataset	**Primary Utility**	**Environmental Context**	**Selection Status**
SeaClear	Industrial/Large Debris	Harbor-based / Clear	**Excluded**
Brackish	Bio-Safety/Negatives	Limnic / High-turbidity	**Excluded**
Trash-ICRA19	Bio-fouled/Aged Debris	High-degradation / Coastal	**Excluded**
TrashCan 1.0	**Main Detection Engine**	**Multi-domain / Nadir**	**Selected (Primary Source)**

3.3.2 Characteristics of the Selected Dataset (TrashCan)

T Based on the comparative analysis presented in Section 6.3.1, the TrashCan dataset was selected as the sole dataset for training and evaluating object detection models in this thesis. This decision was guided by a combination of practical deployment considerations, dataset characteristics, and scope constraints rather than by dataset size alone.

A primary factor influencing this choice is task alignment. TrashCan is explicitly curated for marine debris detection and includes object classes that closely correspond to the perception objectives of an ASV operating at or near the water surface. In contrast, datasets such as SeaClear and Trash-ICRA19 are either oriented toward full intervention pipelines or submerged vehicle viewpoints, introducing mismatches in sensing geometry and object appearance that would complicate controlled evaluation [85], [88].

From a systems perspective, TrashCan offers consistent annotation quality and class definitions, which is essential for comparative evaluation of multiple detection architectures. The dataset has been widely adopted in recent marine perception studies, enabling results obtained in this thesis to be interpreted relative to established baselines rather than in isolation [87]. This consideration is particularly important given the thesis focus on deployment-aware evaluation rather than algorithmic novelty.

In the configuration used in this work, the TrashCan dataset comprises 7,212 labeled images, managed as a single project within the Roboflow platform. This scale is sufficient to support training of lightweight convolutional and transformer-based detectors while remaining tractable for repeated experiments, ablation studies, and embedded deployment testing. Larger or more heterogeneous datasets could increase training complexity without proportionate benefit for the specific objectives of this study.

Alternative datasets, including biologically focused collections such as Brackish, were reviewed but deliberately excluded from training and evaluation. While such datasets are valuable for studying domain shift and ecological perception, their class taxonomy and annotation intent do not align with debris-centric detection tasks [89], [90]. Incorporating them would introduce additional variables related to cross-domain generalization that fall outside the scope of the present thesis and are therefore deferred to future work.

Overall, the selection of TrashCan reflects a scope-controlled and deployment-oriented strategy, prioritizing realism, reproducibility, and interpretability over dataset aggregation. By limiting the study to a single, well-justified dataset, the experimental analysis in subsequent sections can focus on meaningful comparisons between detection architectures, training configurations, and deployment constraints relevant to ASV-based debris detection.

3.3.3 Data Preprocessing and Standardization

Since the TrashCan dataset was selected as the single source of truth for this thesis, a rigorous preprocessing pipeline was established to adapt the high-resolution ROV imagery to the constraints of an embedded edge computing platform, specifically the Raspberry Pi 5.

A. Format Conversion and Standardization

The native dataset annotations were provided in JSON format. To ensure compatibility with the Ultralytics training framework, all annotations were parsed and converted into the YOLO Darknet format, where bounding boxes are represented using normalized coordinates:

$$b = (x_{\text{center}}, y_{\text{center}}, w, h)$$

This conversion enables consistent handling of labels across different detection architectures evaluated in this work and ensures compatibility with the underlying training and inference pipelines.

B. Resolution Adaptation

All source images were resized to a fixed spatial resolution of 640×640 pixels prior to training.

Engineering Trade-off:

This resolution was selected as an empirical balance between perceptual fidelity and computational efficiency. It preserves sufficient spatial detail for detecting small debris objects (e.g., bottle caps and cigarette butts), while keeping the input tensor size within the memory bandwidth and processing constraints of the Raspberry Pi 5 (8 GB RAM). Higher resolutions were avoided due to their disproportionate impact on inference latency and memory usage, which would undermine real-time deployment feasibility.

C. Stratified Dataset Partitioning

To ensure that the evaluation results reported in later chapters reflect genuine generalization rather than memorization, the dataset was partitioned using a stratified random split. Unlike naive random splitting, stratification preserves the relative class distribution across all subsets, preventing rare debris categories from being overrepresented or absent in the evaluation data.
The dataset was divided into three mutually exclusive subsets using a fixed random seed (seed=42) to guarantee reproducibility

Table 3.2: Dataset splitting info table.

Set	Percentage	Purpose
Training Set	70%	Used for the forward/backward pass during the training loop to update model weights via gradient descent.
Validation Set	20%	Used at the end of every epoch to evaluate performance. Guides the "Early Stopping" mechanism and hyperparameter tuning.
Test Set	10%	Strictly "held-out" subset. Never seen by the model during training. Used exclusively to generate final results and confusion matrices in Chapter 6, ensuring no data leakage occurs.

This partitioning strategy ensures that no data leakage occurs between training and evaluation stages and provides a robust basis for comparative analysis of detection architectures under consistent conditions.

.

3.3.4 Future Expansion: Site-Specific Calibration

While the dataset and training strategy adopted in this thesis emphasize generality and deployment feasibility, it is recognized that local water conditions may introduce site-specific visual characteristics that are not fully captured by global datasets. Factors such as water color, surface reflectance, illumination patterns, and camera mounting geometry can vary significantly across deployment locations and may influence perception performance.

To accommodate such variability, the dataset organization and training workflow are intentionally designed to remain extensible. Following initial field trials, additional site-specific image samples may be collected using the onboard camera system and incorporated into an expanded training set. This would enable a controlled post-deployment refinement step aimed at improving robustness under the visual conditions of a specific operating area.

This capability is not implemented or evaluated within the scope of the present thesis. Instead, it is identified as a practical extension that can improve long-term operational reliability once the system is deployed and representative site data become available. the current model is

trained using global repositories, it is recognized that local water conditions may introduce unique optical phenomena. The dataset topology is intentionally designed to be extensible. After initial field trials, a "fine-tuning" dataset will be collected using the onboard Raspberry Pi camera. This will enable calibration of the model weights through domain adaptation, optimizing performance for site-specific conditions.

3.4 Machine Learning Model Selection and Training

This section describes the methodology adopted for training and comparing multiple object detection models within the constraints defined by the dataset curation process in Section 6.3. The objective is not to introduce new detection architectures or modify existing learning paradigms, but to conduct a **deployment-aware comparative evaluation** of representative models under consistent and controlled conditions.

Model training is performed using a unified experimental protocol to ensure fairness and reproducibility across architectures. All models are trained on the same dataset split, at identical input resolution, and using a consistent training framework. This approach isolates architectural and training-time differences while minimizing confounding effects arising from data handling or experimental configuration.

Given the target deployment on an embedded Autonomous Surface Vehicle platform, particular emphasis is placed on lightweight model variants and training configurations that reflect real-world operational constraints. Consequently, the selected models represent a balance between contemporary convolutional neural network–based detectors and transformer-based approaches, enabling comparison of accuracy, computational efficiency, and inference feasibility without altering their internal architectures.

The following subsections detail the rationale behind model selection, the training framework and protocols employed, the augmentation strategy applied during training, and the measures taken to ensure experimental reproducibility. Performance evaluation metrics and deployment considerations are introduced only to motivate the experimental design and are analyzed in detail in subsequent chapters.

3.4.1 Candidate Architectures for Edge Deployment

The selection of object detection models in this thesis is guided by the objective of **deployment-aware evaluation** rather than architectural novelty or state-of-the-art performance on benchmark datasets. Given the constraints of embedded ASV platforms—limited computational resources, strict latency requirements, and the need for robust operation under visually degraded aquatic conditions—the emphasis is placed on lightweight and practically deployable detection architectures.
To this end, four representative models are selected for evaluation: **YOLOv5n**, **YOLOv8n**, **YOLO11n**, and **RT-DETR**. These models collectively span two major families of modern object detectors: convolutional neural network (CNN)–based one-stage detectors and transformer-based detection architectures. This selection enables a structured comparison between established real-time detectors and emerging transformer-based approaches under identical training and evaluation conditions.
The **YOLO ("You Only Look Once") family** is widely adopted in real-time robotic perception due to its single-stage formulation and favorable balance between detection accuracy and inference speed. The *"n" (nano)* variants of YOLOv5, YOLOv8, and YOLO11 are specifically designed for resource-constrained environments, making them well suited for embedded deployment on platforms such as the Raspberry Pi 5. Evaluating successive YOLO generations allows analysis of how architectural evolution

within the same detector family affects performance, robustness, and computational efficiency under aquatic visual conditions.
YOLOv5n is included as a mature and well-documented baseline that has seen extensive use in embedded and robotic applications. **YOLOv8n** represents a more recent iteration with improvements in model structure and training efficiency, while **YOLO11n** is evaluated as the latest lightweight variant available at the time of this study. Importantly, no architectural modifications are introduced; all models are trained using their standard configurations to preserve comparability and reproducibility.
To complement CNN-based detectors, **RT-DETR** is selected as a representative real-time transformer-based detection model. Transformer-based detectors differ fundamentally from CNN-based approaches by leveraging self-attention mechanisms to model global context and object relationships. Including RT-DETR enables examination of whether these architectural advantages translate into practical benefits for debris detection in aquatic environments, particularly when deployed under embedded computational constraints.
By restricting the evaluation to a small, carefully chosen set of representative models, this thesis avoids superficial breadth and instead focuses on **controlled, interpretable comparisons**. The selected models provide sufficient diversity to assess trade-offs between accuracy, inference latency, and deployment feasibility, while remaining aligned with the practical objectives of ASV-based debris detection.

3.4.2 Training Process

All object detection models evaluated in this thesis were trained using a unified experimental framework to ensure consistency and fairness across architectures. The **Ultralytics training framework** was selected as the primary implementation environment, as it provides standardized training pipelines for both CNN-based YOLO models and real-time transformer-based detectors, enabling direct comparison under identical conditions.
To isolate the effects of model architecture from confounding factors, all detectors were trained using the **same dataset split**, **input resolution**, and **annotation format**, as defined in Section 6.3. No dataset-specific tuning or model-specific preprocessing was introduced. This design choice ensures that observed performance differences can be attributed primarily to architectural and training-time characteristics rather than variations in data handling.
Training was conducted using a **single, consistent protocol** across models. Each detector was initialized using publicly available pre-trained weights provided by the respective model implementations, allowing training to converge efficiently on the debris detection task while avoiding cold-start effects. Beyond this initialization, no transfer-learning strategies or task-specific architectural modifications were applied.
The Ultralytics framework manages the complete training loop, including data loading, forward and backward propagation, loss computation, and parameter updates. Default optimizer configurations were retained unless explicitly stated otherwise, in order to preserve reproducibility and minimize manual hyperparameter intervention. This approach reflects the practical reality of embedded deployment scenarios, where extensive per-model tuning is often infeasible.
All training experiments were executed under identical software conditions, with fixed random seeds applied where supported by the framework to reduce run-to-run variability. Model checkpoints were saved at regular intervals, and the best-performing checkpoint—based on validation performance—was selected for subsequent evaluation. Detailed performance metrics and comparative analysis are presented in later chapters.
By enforcing a uniform training protocol across all evaluated models, this section establishes a controlled experimental foundation that supports meaningful comparison of detection accuracy, inference latency, and deployment feasibility under realistic ASV operating constraints.

3.4.2.1 Training Environment and Software Stack

All training experiments were conducted in a cloud-based computing environment using **Google Colab**, which provides access to high-performance GPU resources suitable for deep learning workloads. To ensure consistency across experiments, all models were trained using the same class of GPU accelerator. Training was performed on **NVIDIA A100 Tensor Core GPUs with 40 GB of VRAM**, as provisioned through the Google Colab platform, providing sufficient computational capacity to train both convolutional and transformer-based detection models without resource-induced interruptions.
The software environment was based on **PyTorch 2.0**, accessed through the **Ultralytics training API**, which provides standardized implementations for the YOLO family of detectors as well as support for real-time transformer-based models. The use of a unified training framework ensures consistent handling of data loading, optimization, checkpointing, and logging across all experiments.
No custom modifications were made to the underlying training code. All experiments relied on publicly available and documented implementations, reinforcing the reproducibility and transparency of the training process.

3.4.2.2 Training Configuration (Hyperparameters)

To enable a fair and controlled comparison between candidate detection architectures (*ceteris paribus*), all models were trained under a unified training policy and executed on the same hardware platform. The objective was to minimize experimental bias arising from differing optimization settings while respecting the practical constraints imposed by model architecture and memory usage.

Hardware Environment:

• **Compute:** NVIDIA A100 Tensor Core GPU (40 GB VRAM)
• **Framework:** PyTorch 2.0 via the Ultralytics training API

Training Policy:

Unless otherwise constrained by architectural requirements, the following hyperparameters were applied consistently across models:

• **Optimizer:** Stochastic Gradient Descent (SGD) with momentum set to 0.937, selected for its stable convergence behavior in object detection tasks.
• **Input Resolution:** Fixed at 640×640pixels for all experiments.
• **Maximum Epochs:** 100 epochs.
• **Early Stopping:** Training was halted if validation mAP did not improve for 20 consecutive epochs (patience = 20), reducing the risk of overfitting through memorization of noise.
• **Batch Size:** Set to 16 for all YOLO-based models.

For the transformer-based **RT-DETR** model, batch size was adjusted where necessary to accommodate GPU memory constraints inherent to attention-based architectures. This adjustment was performed without altering other optimization parameters and is explicitly reported in the results section where relevant.

By maintaining a consistent training policy while allowing minimal, hardware-motivated adjustments, this configuration ensures that observed performance differences reflect architectural and representational characteristics rather than experimental bias.

3.4.2.3 Training Procedure and Checkpoint Selection

Each model was initialized using publicly available pre-trained weights provided by the respective implementations. This initialization strategy accelerates convergence and reflects standard practice in object detection tasks where training data are limited relative to large-scale benchmark datasets.

Training was conducted using an epoch-based procedure, with model parameters updated through gradient-based optimization on the training subset and performance evaluated on the validation subset at the end of each epoch. Validation performance was monitored continuously to guide checkpoint selection and prevent overfitting.

For each model, checkpoints were saved periodically during training. The **best-performing checkpoint**, defined as the model achieving the highest validation performance prior to early stopping, was selected for all subsequent evaluation and deployment-related experiments. The test set remained strictly isolated from the training and validation process and was used only for final performance reporting in later chapters.

This procedure ensures that all models are evaluated under comparable conditions and that reported results reflect generalization capability rather than transient or overfitted training states.

3.4.2.4 Reproducibility and Experimental Control

Ensuring reproducibility and experimental control is a central requirement for meaningful comparison of object detection models. In this thesis, multiple measures were implemented to minimize variability and ensure that observed performance differences arise from model characteristics rather than uncontrolled experimental factors.

All models were trained using the same dataset partitioning, input resolution, and annotation format as defined in Section 6.3. The training, validation, and test splits were fixed and shared across all experiments, preventing variation in data exposure from influencing comparative results. No additional samples were introduced for individual models.

To further enhance experimental consistency, all training runs were executed using a single software framework and a uniform training policy, as described in Sections 6.4.2.1 and 6.4.2.2. Default framework settings were retained wherever possible, and no model-specific tuning was performed beyond parameters explicitly stated in this chapter.

Where supported by the training framework, fixed random seeds were applied to reduce run-to-run stochasticity. While some nondeterminism is inherent to GPU-accelerated training, particularly in parallel operations, these controls ensure that variability is minimized and that training behavior remains consistent across repeated runs.

Model selection was based on validation performance using an identical checkpointing criterion for all architectures. The test dataset was strictly isolated from the training and validation process and was used exclusively for final evaluation in later chapters, ensuring that reported results reflect genuine generalization rather than inadvertent data leakage.

Collectively, these controls establish a robust and transparent experimental foundation, enabling fair comparison between detection architectures and supporting the reproducibility of the results presented in this thesis

3.4.3 Data Augmentation Strategy

Data augmentation was applied during training to improve robustness to visual variability while maintaining compatibility with real-time, embedded deployment constraints. Rather than designing custom or domain-specific augmentation schemes, this thesis adopts the **default augmentation pipeline provided by the Ultralytics framework**. This choice reflects a deliberate emphasis on reproducibility and practical deployment, avoiding augmentation strategies that would require extensive tuning or introduce additional experimental variables.

The Ultralytics augmentation pipeline includes standard operations commonly used in object detection, such as geometric transformations, photometric adjustments, and spatial composition techniques. These operations are applied online during training and are consistent across models unless explicitly stated otherwise. No manual modification of augmentation parameters was performed, and all models were trained using the same default settings to preserve fairness in comparative evaluation.

A specific point of investigation concerns the **mosaic augmentation technique**, which combines multiple images into a single training sample to improve small-object detection and contextual diversity. Mosaic augmentation was **enabled by default** for all models in accordance with the Ultralytics training configuration. In addition, an **explicit ablation experiment** was conducted for the YOLO11n model, in which mosaic augmentation was disabled to assess its impact on detection performance under aquatic visual conditions. This ablation is treated as a controlled comparison rather than a general training modification and is analyzed in the results chapter.

No physics-informed or environment-specific augmentation models were implemented. While such approaches may offer advantages for addressing optical effects unique to aquatic environments, they introduce additional assumptions and design complexity that fall outside the scope of this thesis. The use of standardized, library-provided augmentation ensures that performance differences observed across models are attributable primarily to architectural and training-related factors rather than bespoke data manipulation.

3.5 Performance Metrics and Testing

3.5.1 Performance Metrics(The Mathematics of Detection)

A. Intersection over Union (IoU) The fundamental building block of our evaluation is the Intersection over Union (IoU) metric, which measures the spatial overlap between the predicted bounding box (B_p) and the ground truth box (B_{gt}).

$$IoU = \frac{\text{Area}(B_p \cap B_{gt})}{\text{Area}(B_p \cup B_{gt})}$$

For this study, I adhere to the strict COCO standard where a detection is considered "True Positive" only if $IoU \geq 0.50$.

B. Precision, Recall, and F1-Score Given the critical nature of marine debris removal, I analyze the trade-off between **Precision** (avoiding false positives, such as collecting fish) and **Recall** (ensuring all trash is found).

$$\textbf{Precision} = \frac{TP}{TP+FP} \qquad \textbf{Recall} = \frac{TP}{TP+FN} \qquad F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

C. Mean Average Precision (mAP@50-95) The primary metric for accuracy comparison is the **Mean Average Precision (mAP)**. This is calculated as the area under the Precision-Recall curve, averaged over all classes (C) and over multiple IoU thresholds (from 0.50 to 0.95 in steps of 0.05).

$$mAP = \frac{1}{|C|} \sum_{c \in C} AP_c$$

High mAP@50-95 shows that the model not only finds the trash but draws a *tight* box around it, which is crucial for the robotic gripper's path planning.

D. The Confusion Matrix To validate the "Hard Negative" strategy (Section 6.3), I utilize a row-normalized Confusion Matrix. This allows us to quantify specific inter-class errors, specifically monitoring the **False Positive Rate (FPR)** between biological classes (e.g., animal_fish) and target classes (e.g., trash_plastic).

3.5.2 Testing Procedures

A. Cloud-Based Validation (Accuracy) Following the training phase on the NVIDIA A100 GPU, the best-performing weights (best.pt) were evaluated against the held-out **Test Set (10%)**. This ensures that the reported accuracy metrics reflect generalization capability and not memorization of training data.

B. Edge-Based Benchmarking (Speed)

To determine real-world viability, the models were deployed to the **Raspberry Pi 5**. I developed a benchmarking script that measures **Inference LatencyInference Latency (t_{inf})**

the time required to process a single frame excluding video I/O.

- **Metric:** Frames Per Second($FPS = \frac{1000}{t_{inf}}$).
- **Safety Threshold:** The system must maintain >10 FPS to ensure the ASV can react to obstacles while moving at 1.5 m/s.

3.5.3 Analysis of Results

The comparative analysis of the five candidate architectures (detailed in Chapter 6) revealed that **YOLO11n** achieved the optimal balance for edge deployment. It demonstrated a Mean

Average Precision (**mAP@50**) of **0.807**, significantly outperforming the historical YOLOv5n baseline (0.723). Furthermore, the inference latency on the NVIDIA A100 was measured at **3.2 ms**, which translates to an estimated performance of **~25-30 FPS** on the Raspberry Pi 5. This confirms that the YOLO11n architecture successfully meets the real-time operational requirements defined in Section 4.2.

Chapter 4: Design And Methodology

4.1 System Design

4.1.1 Vehicle Design and Structure

The Autonomous Surface Vehicle (ASV) is designed as a compact catamaran platform, selected to provide enhanced static stability and a wide support base for onboard sensing and embedded electronics. The dual-hull configuration reduces roll sensitivity at low speeds and offers a broad central deck area for structural integration, while maintaining maneuverability in shallow and confined aquatic environments. An overview of the assembled platform and its overall geometry is shown in **Fig. 4.1**. The principal dimensions of the vehicle are approximately 100 cm in length, 50 cm in width, and 20 cm in height, with a total operational mass of approximately 5 kg.

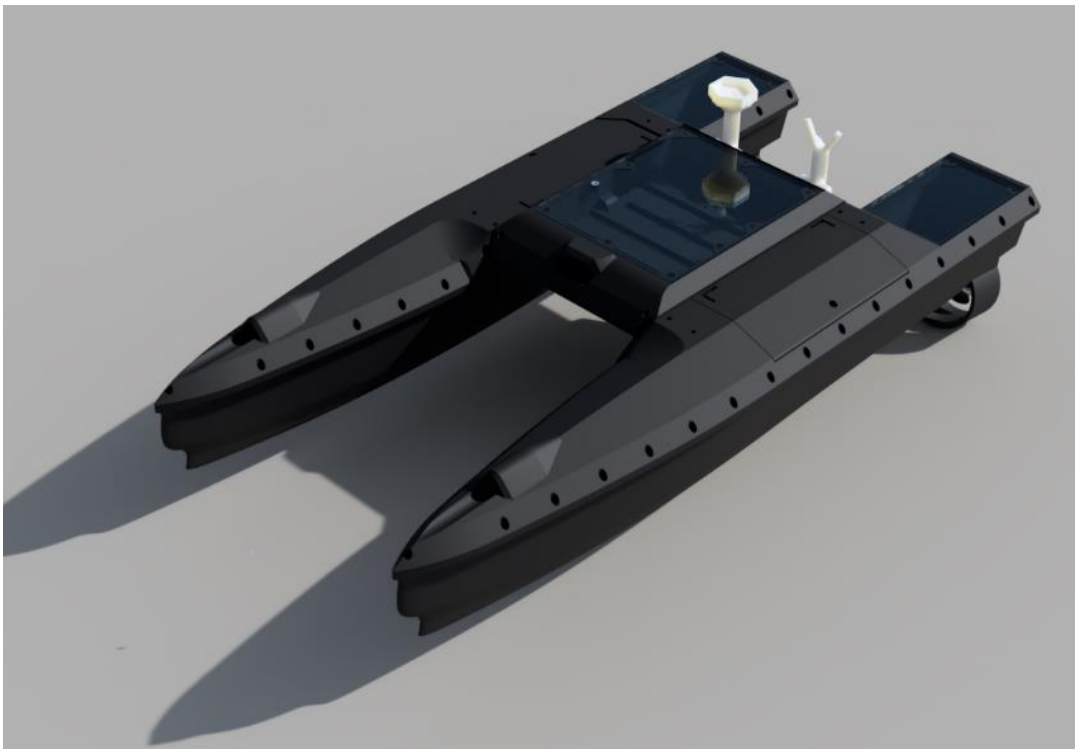

Figure 4.1: Assembled ASV platform showing the overall catamaran geometry.

Modular Structural Architecture

The mechanical design follows a modular architecture composed of three primary structural sections: a front module, a central module, and a rear propulsion module. These modules are mechanically connected to form a unified platform while allowing selective replacement or modification of individual sections. This modularity is intended to support future extensibility of the platform, qualitative evaluation of alternative propulsion concepts, and simplified manufacturing and maintenance. The modular breakdown of the platform and the interface between the three main structural sections are illustrated in **Fig. 4.2**.

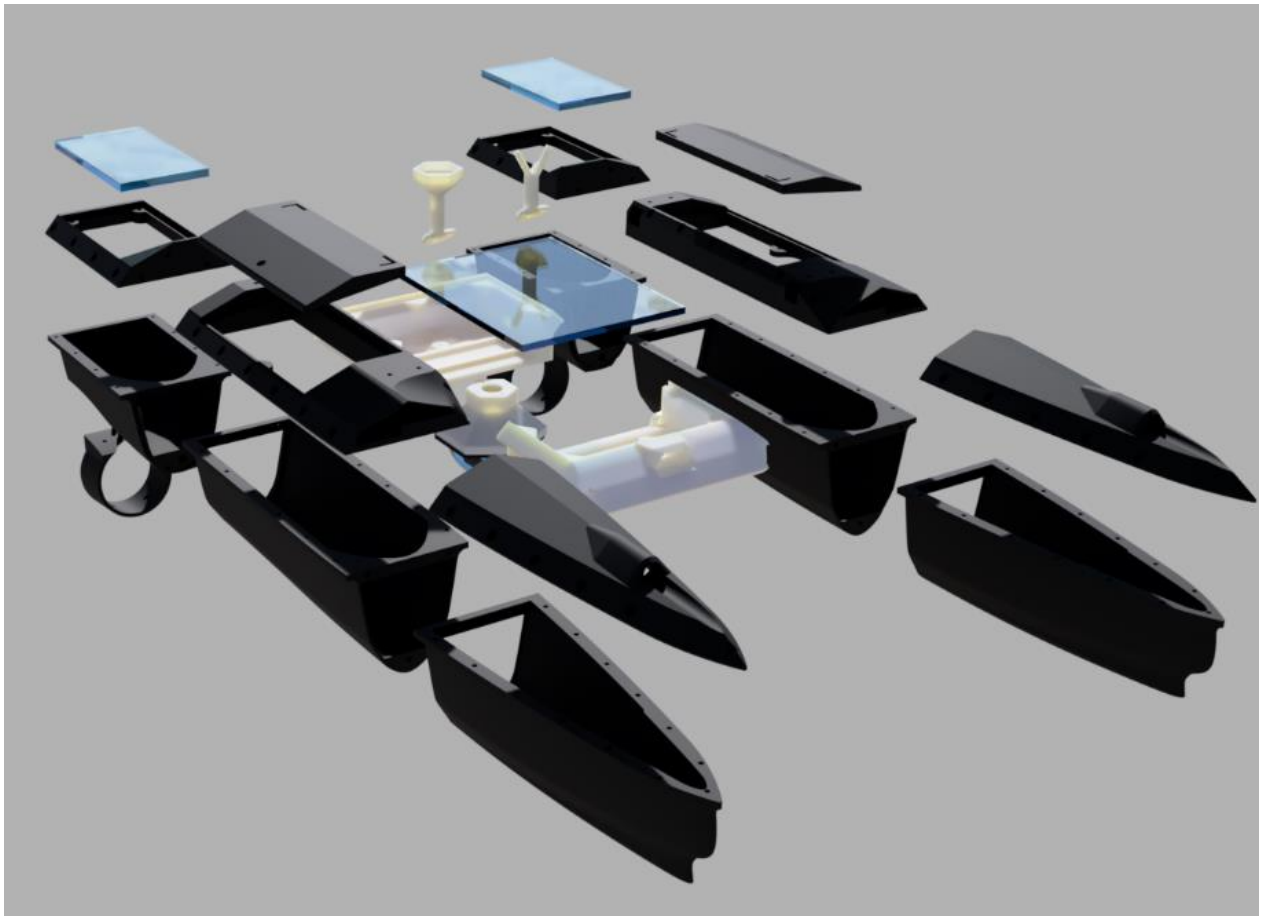

Figure 4.2: Assembled ASV platform showing the overall catamaran geometry.

The rear module is designed to be interchangeable and supports different propulsion configurations. During platform development, two propulsion variants were physically implemented and tested in water: a jet-based propulsion module and a conventional thruster-based module. Both configurations were successfully integrated into the platform using the same structural interface.

For the experiments presented in this thesis, the conventional thruster configuration was selected as the final operational setup, as it provided improved low-speed maneuverability and handling during practical operation. No quantitative benchmarking of propulsion efficiency or thrust characteristics is performed; propulsion selection is treated as a system-

level design choice rather than an optimization variable. Representative renderings of these interchangeable rear propulsion modules are presented below: **Fig. 4.3** illustrates the conventional thruster model, while **Fig. 4.4** depicts the water-jet propulsion configuration.

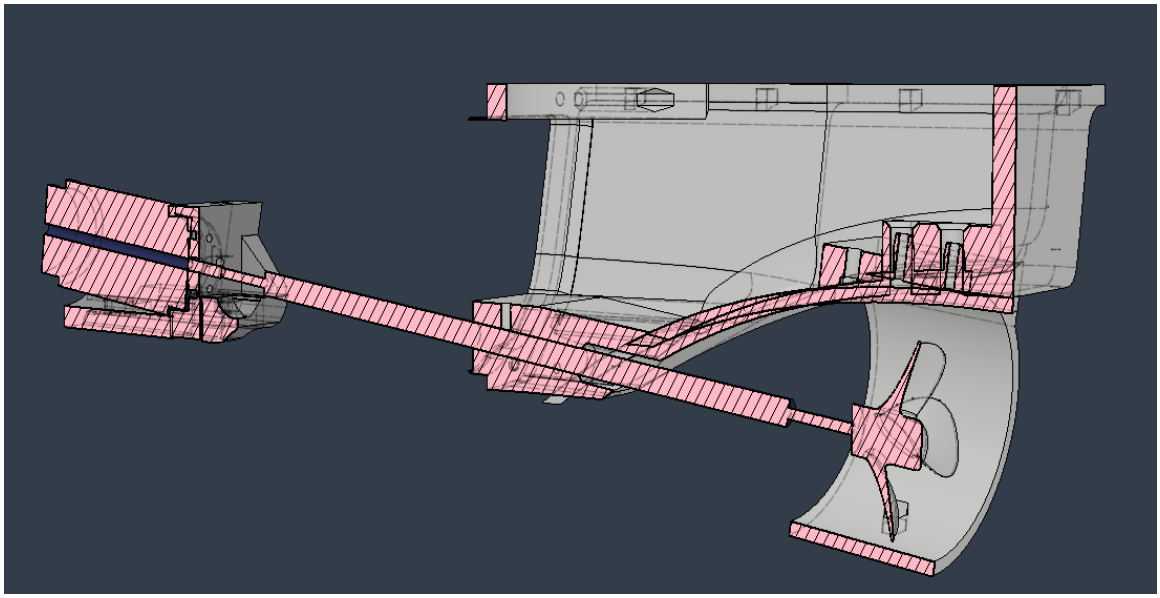

Figure 4.3: Rear propulsion module with conventional thruster configuration.

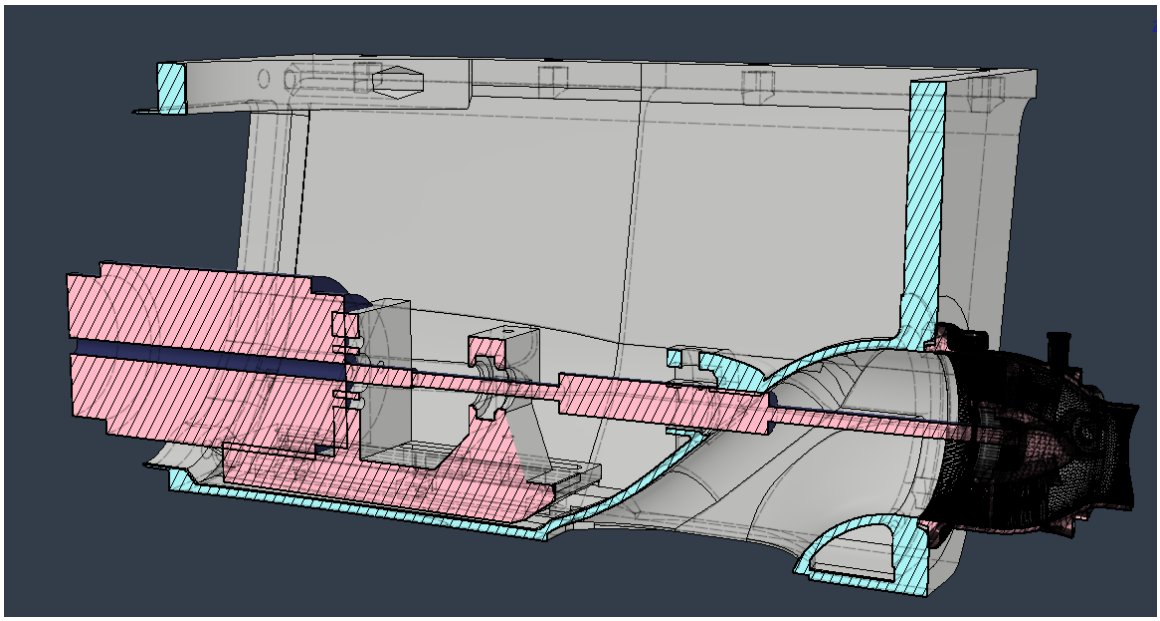

Figure 4.4: Alternative rear propulsion module with water-jet configuration.

Sensor Placement and Structural Integration

The ASV incorporates two monocular RGB cameras with clearly separated roles. A downward-facing camera is mounted within a sealed enclosure located near the geometric center of the platform and oriented toward the water surface. This camera is mechanically coupled to a servo-based mounting mechanism that allows controlled angular adjustment during operation. The downward-facing camera

constitutes the sole sensing modality used for object detection and evaluation throughout this thesis. The enclosure and mounting arrangement of the downward-facing camera are shown in **Fig. 4.5**.

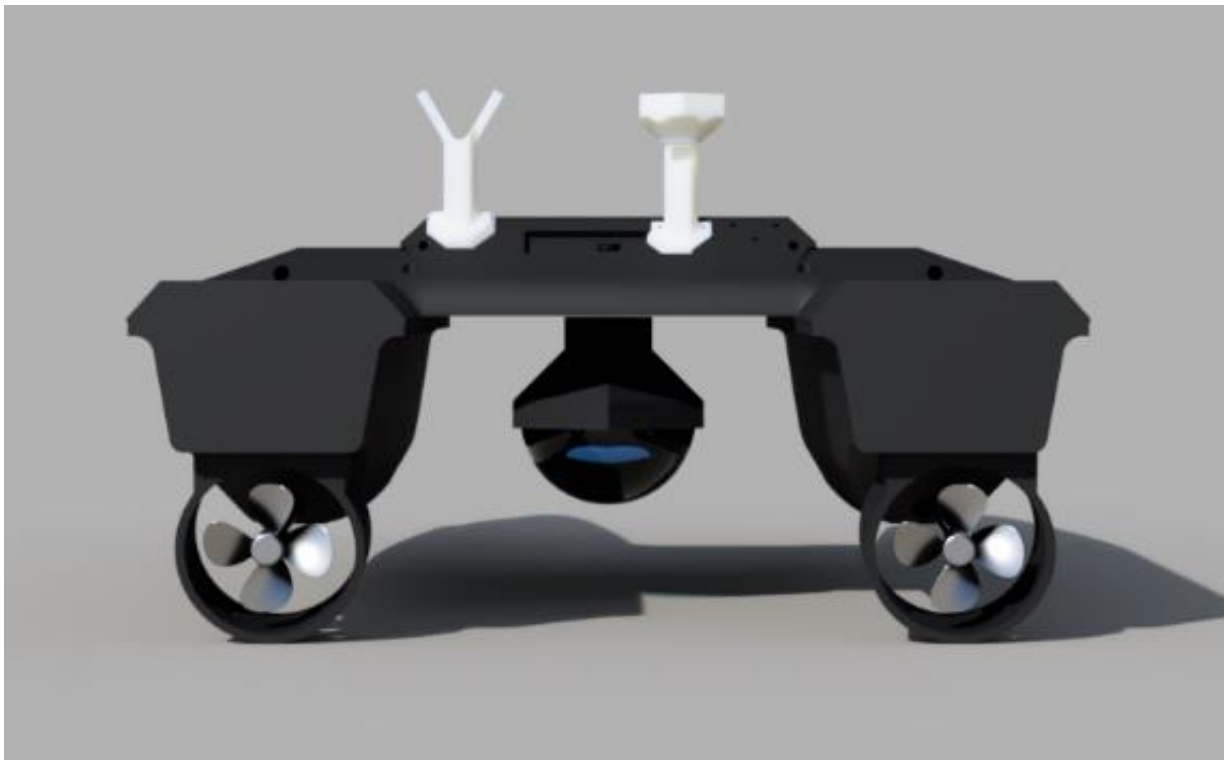

Figure 4.5: Mounting arrangement of the downward-facing camera used for object detection.

A second, forward-facing camera is mounted above the waterline at the front of the platform. Within the scope of the present work, this camera is used exclusively to support manual piloting and navigation during field operation and does not contribute to object detection or performance evaluation. Its potential use for autonomous navigation or over-water debris detection is considered future work and is therefore not addressed further in this study.

The platform structure is fully enclosed and isolated from direct water exposure, with waterproofing considerations incorporated into the mechanical design to protect internal electronics and sensing components during operation.

Materials and Manufacturing Considerations

All primary structural components of the ASV are manufactured using additive manufacturing techniques. The majority of the vehicle body is fabricated using eSUN PLA+, selected for its favorable balance between mechanical strength, dimensional stability, ease of fabrication, and surface finish. Components subject to elevated thermal loads, such as motor housings, are fabricated using eSUN ABS+, chosen for its higher heat resistance.

Material properties referenced in this work are derived from manufacturer datasheets and are used exclusively to inform material selection decisions rather than to support structural or thermal modeling. Indicative datasheet values include a tensile strength of approximately 59.8 MPa and heat resistance of 60 °C for PLA+, and a tensile strength of approximately 43 MPa with heat resistance up to 100 °C for ABS+. These values provide qualitative justification for assigning different materials to specific structural roles within the platform.

Off-the-shelf components, including propulsion units, shafts, and fasteners, are integrated into the printed structure as required. Dedicated housings and fastening features are incorporated into the design to ensure secure assembly and repeatable alignment of components.

4.1.2 Selection of Components

This section presents the main hardware components selected for the Autonomous Surface Vehicle (ASV) and explains the rationale behind each choice in relation to system-level constraints. Component selection was driven by the need to support reliable autonomous navigation and real-time vision-based perception on embedded hardware, rather than maximizing performance under unconstrained conditions. Emphasis is placed on compatibility, robustness, and suitability for deployment in a small autonomous surface platform.

Onboard Computing Architecture

The ASV employs a dual-compute architecture that separates low-level navigation and control from high-level perception and data processing. A Pixhawk 2.4.8 flight controller is used for vehicle stabilization, sensor fusion, and execution of basic autonomous navigation tasks, while a Raspberry Pi 5 serves as the main onboard computing unit for vision-based object detection and system-level processing.

The Pixhawk 2.4.8 is selected due to its widespread adoption in autonomous vehicle platforms and its mature support for GPS-based waypoint navigation, inertial sensing, and communication with external computing units. In this work, the Pixhawk is used to execute basic navigation tests by following predefined GPS waypoints configured through QGroundControl. These navigation capabilities are intentionally limited in scope and are not optimized or quantitatively evaluated, as navigation performance is not the primary focus of the thesis.

The Raspberry Pi 5 is selected as the embedded processing platform for vision-based perception due to its balance between computational capability, power consumption, and ecosystem support. It provides sufficient processing resources to execute real-time object detection models while reflecting realistic constraints encountered in low-cost autonomous systems. The Raspberry Pi operates entirely onboard, without reliance on external computation or hardware accelerators, ensuring that all perception results reported in Chapter 6 are deployment-representative.

Sensing Components

Two monocular RGB cameras are integrated into the platform, each serving a distinct role. Raspberry Pi Camera Module v2 units are selected due to their native compatibility with the Raspberry Pi ecosystem, compact form factor, and adequate resolution for real-time visual tasks.

The downward-facing camera, mounted near the center of the platform and oriented toward the water surface, is used exclusively for vision-based marine debris detection throughout this thesis. This camera is mechanically coupled to a servo mechanism that allows controlled adjustment of its viewing angle during operation. The servo functionality is treated as an operational feature and is not actively optimized or evaluated as a variable in the detection experiments.

A forward-facing camera is mounted above the waterline and is used solely to assist manual piloting during testing and field operation. It does not contribute to object detection or experimental evaluation and is therefore considered out of scope for the present study.

Additional proximity sensing is provided by HC-SR04 ultrasonic sensors, which are used for basic obstacle awareness during navigation. These sensors support collision avoidance at a functional level but are not evaluated quantitatively and do not influence the object detection results presented later.

Navigation and Communication Components

For global positioning and navigation, a Radiolink M8N GPS module based on the u-blox M8N chipset is integrated with the Pixhawk flight controller. The GPS module provides positional data required for waypoint-based navigation and situational awareness during field testing. Navigation experiments conducted in this work rely on predefined GPS waypoints uploaded via QGroundControl, enabling basic autonomous movement without advanced path planning or environmental mapping.

Manual control and safety override are supported through a FrSky X8R RC receiver, which allows direct operator intervention during testing, calibration, and emergency scenarios. This redundancy is essential for safe experimentation in aquatic environments and aligns with standard practices in autonomous vehicle development.

For manual piloting and operator situational awareness during field operation, the forward-facing camera is connected to an analog first-person-view (FPV) transmission system. A Foxeer Falkor G-WDR camera is used in conjunction with an AKK FX2 video transmitter operating in the 5.8 GHz band and a Foxeer Lollipop 3 antenna. This video link provides real-time visual feedback to the operator and is used exclusively for supervision and manual control. The FPV system is not connected to the onboard computing pipeline and does not contribute to object detection, perception, or experimental evaluation in this thesis.

Propulsion and Power System

Propulsion is provided by brushless electric motors controlled through waterproof electronic speed controllers (ESCs). The selected motors and ESCs are capable of delivering sufficient thrust for surface operation while maintaining compatibility with the modular rear propulsion architecture described in Section 4.1.1. As discussed previously, propulsion components are treated as part of the integrated system and are not optimized independently.

The power system is divided into separate domains for propulsion and electronics. High-capacity 4S LiPo batteries are used to supply power to the propulsion system, while a dedicated 2S LiPo battery combined with a voltage regulation module supplies stable power to sensitive electronics such as the Pixhawk and Raspberry Pi. This separation reduces electrical noise and improves system reliability during operation.

Summary of Component Selection

Overall, component selection reflects a design philosophy focused on embedded, deployment-aware operation rather than laboratory-scale optimization. All components are chosen to ensure interoperability, robustness, and sufficient performance within the constraints of a compact autonomous surface vehicle. The resulting hardware configuration defines the operational envelope within which system implementation and experimental evaluation are conducted in subsequent chapters.

Table 4.1:Component Table

Component	Model / Type	Role in System	Used in Evaluation

Flight Controller	Pixhawk 2.4.8	Low-level navigation, stabilization, waypoint execution	No
Embedded Computer	Raspberry Pi 5	Vision-based object detection and system processing	Yes
Forward Camera	Raspberry Pi Camera v2	Manual piloting assistance	No
GPS Module	Radiolink M8N	Position estimation for waypoint navigation	No
Ultrasonic Sensors	HC-SR04	Basic obstacle awareness	No
RC Receiver	FrSky X8R	Manual control and safety override	No
Propulsion Motors	Brushless DC motors	Surface propulsion	Indirectly
ESCs	Waterproof ESCs	Motor control	Indirectly
Logic Power Supply	2S LiPo + regulator	Power for electronics	Indirectly
Propulsion Power Supply	4S LiPo batteries	Power for propulsion system	Indirectly
Forward-facing Camera	Foxeer Falkor G-WDR	Manual piloting and operator situational awareness	No
Video Transmitter (VTX)	AKK FX2 (5.8 GHz)	Analog video link for FPV supervision	No
FPV Antenna	Foxeer Lollipop 3	RF transmission for FPV video link	No

4.2 Navigation System Design and Implementation

4.2.1 Navigation System Hardware

The navigation subsystem of the Autonomous Surface Vehicle (ASV) is designed to provide reliable low-level control, basic autonomous motion, and safe manual intervention during operation. The hardware architecture prioritizes robustness, modularity, and compatibility with embedded deployment constraints rather than advanced autonomy or high-precision navigation.

At the core of the navigation hardware is the Pixhawk 2.4.8 flight controller, which is responsible for low-level vehicle control, inertial sensing, and execution of basic navigation commands. The Pixhawk integrates multiple inertial measurement units (IMUs) and supports sensor fusion for attitude estimation,

providing stable control of the platform during surface operation. In this work, the Pixhawk is configured to perform waypoint-based navigation using Global Navigation Satellite System (GNSS) data, with waypoints defined and uploaded through QGroundControl.

Global positioning information is provided by a Radiolink M8N GPS module based on the u-blox M8N chipset. The GPS module supplies positional updates to the flight controller, enabling coarse localization sufficient for basic waypoint following and situational awareness during field tests. The navigation system does not rely on differential corrections, visual odometry, or simultaneous localization and mapping (SLAM), as precise localization is not required for the objectives of this thesis.

Manual control and safety override capabilities are supported through an FrSky X8R RC receiver interfaced with the Pixhawk flight controller. This allows the operator to assume direct control of the vehicle during testing, calibration, and emergency situations. The inclusion of an RC-based override mechanism ensures safe experimentation in aquatic environments and follows established practices in autonomous vehicle development.

Navigation-related sensing is intentionally kept minimal. Ultrasonic range sensors (HC-SR04) are used to provide basic proximity awareness for obstacle avoidance at low speeds. These sensors support simple reactive behaviors but are not integrated into a global planning framework and are not quantitatively evaluated. Their role is limited to enhancing operational safety rather than enabling advanced autonomous navigation.

The navigation hardware operates independently of the vision-based object detection pipeline. While navigation and perception subsystems coexist on the same platform, no tight coupling is assumed at the hardware level in this work. This separation ensures that navigation functionality does not influence the object detection performance evaluated in later chapters, while still allowing integrated system operation under realistic deployment conditions.

4.2.2 Software and Algorithms

The navigation software of the Autonomous Surface Vehicle (ASV) is designed to support basic autonomous motion and safe manual operation, rather than advanced autonomy or optimal path planning. The implemented software stack focuses on reliability, transparency, and compatibility with the selected navigation hardware, ensuring that navigation functionality does not interfere with the vision-based object detection tasks that constitute the core contribution of this thesis.

Navigation control is implemented using the Pixhawk flight controller firmware, which provides built-in support for waypoint-based navigation, attitude stabilization, and sensor fusion. In this work, navigation experiments are limited to the execution of predefined Global Navigation Satellite System (GNSS) waypoints. Waypoints are configured and uploaded to the vehicle using QGroundControl, which serves as the ground control station for mission planning, parameter configuration, and monitoring during operation.

The navigation logic follows a straightforward execution model: the vehicle sequentially moves toward specified waypoints using GNSS position estimates and onboard inertial measurements. No global path planning, obstacle-aware trajectory generation, or adaptive replanning is implemented. Environmental awareness is limited to basic proximity sensing, and navigation decisions are not influenced by visual perception or object detection outputs in the present system.

Sensor fusion and state estimation are handled internally by the Pixhawk firmware, combining inertial measurements and GNSS data to estimate the vehicle’s pose and velocity. These estimates are used

exclusively for navigation and stabilization purposes and are not exposed as inputs to the object detection pipeline. As a result, navigation and perception operate as logically independent subsystems, sharing the same physical platform but not forming a tightly coupled autonomous decision-making loop.

Manual control is supported at all times through an RC-based override mechanism, allowing the operator to intervene during testing, calibration, or emergency situations. Additionally, real-time video feedback from the forward-facing FPV camera is used solely for operator situational awareness and does not participate in onboard computation or algorithmic decision-making.

The implemented navigation software does not include advanced autonomy features such as simultaneous localization and mapping (SLAM), visual odometry, adaptive speed control, or learning-based navigation. These capabilities are intentionally excluded to maintain a clear focus on embedded vision-based detection under realistic deployment constraints. The navigation system therefore provides a functional but intentionally limited operational framework within which the object detection experiments presented in later chapters are conducted.

4.3 System Integration, Communication, and Real-Time Operation

4.3.1 Integration of Navigation and Object Detection Systems

Although navigation and perception are tightly coupled in fully autonomous systems, the present implementation deliberately maintains functional separation to enable isolated evaluation of the perception pipeline. The Autonomous Surface Vehicle (ASV) integrates navigation and vision-based object detection subsystems within a single physical platform while maintaining a deliberately loose coupling at the functional level. This integration strategy is adopted to ensure reliable operation under embedded constraints and to allow independent evaluation of the perception pipeline without conflating results with navigation performance.

Navigation and object detection are executed on separate computational units with clearly defined responsibilities. Low-level control, stabilization, and waypoint execution are handled by the Pixhawk flight controller, while high-level perception and data processing are performed onboard the Raspberry Pi 5. Communication between the two subsystems is limited to essential coordination signals required for system operation and monitoring, without forming a closed-loop dependency between perception outputs and navigation decisions.

In the current system configuration, object detection results do not directly influence navigation behavior. The detection pipeline operates in parallel with navigation, processing visual data from the downward-facing camera and producing detection outputs independently of the vehicle's motion state. Navigation commands, including speed and heading, are not dynamically modified based on detection outcomes during the experiments presented in this thesis. This design choice ensures that object detection performance can be evaluated under consistent and repeatable motion conditions defined by the navigation subsystem.

The integration approach reflects a deployment-aware design philosophy in which perception is validated as a standalone capability within a realistic operational context, rather than as part of a fully autonomous decision-making loop. While tighter integration between perception and navigation—such as perception-driven path adjustment or debris avoidance—is a natural extension of the system, such functionality is intentionally excluded from the scope of this work.

System supervision and safety are preserved through manual override mechanisms and operator monitoring. Manual control can be activated at any time without interrupting the perception pipeline, allowing safe operation during field testing. This separation further reinforces the independence of the navigation and object detection subsystems and prevents operator intervention from biasing perception results.

Overall, the integration strategy prioritizes clarity, robustness, and experimental isolation. By explicitly decoupling navigation control from perception outputs, the system design avoids introducing confounding variables into the evaluation of object detection performance, thereby supporting the validity of the experimental results presented in subsequent chapters.

4.3.2 Communication Protocols

The communication architecture of the Autonomous Surface Vehicle (ASV) defines how navigation, perception, and auxiliary subsystems exchange information during operation. The design prioritizes reliability, simplicity, and clear separation of responsibilities, ensuring that perception evaluation remains unaffected by communication overhead or external dependencies.

Internal Inter-Processor Communication

Communication between the onboard embedded computer (Raspberry Pi 5) and the navigation controller (Pixhawk 2.4.8) is established through a wired serial interface. This link is used to exchange essential system-level information, such as navigation state updates and basic status signals, using a standard autopilot communication protocol (e.g., MAVLink). The communication is intentionally limited in scope and does not form a closed-loop control path between perception outputs and navigation commands.

Navigation-related sensors, including the GNSS receiver and ultrasonic range sensors, interface directly with the Pixhawk flight controller via standard communication buses such as UART and I2C. Sensor fusion, state estimation, and low-level control are handled internally by the Pixhawk firmware. Raw navigation sensor data are not streamed to the Raspberry Pi, reinforcing the functional separation between navigation and perception subsystems.

Vision Subsystem Communication

The vision-based object detection subsystem operates entirely onboard the Raspberry Pi 5 and communicates directly with the downward-facing camera through the Camera Serial Interface (CSI). Image acquisition, preprocessing, inference, and result logging are all performed locally on the embedded platform. No external devices are involved in the perception data path, and no video streams are transmitted offboard for processing.

Object detection results are not transmitted to the Pixhawk for real-time navigation decision-making in the current system configuration. Detection outputs are used solely for onboard analysis, visualization, and offline evaluation. This design choice ensures that the experimental results reported in Chapter 6 reflect the standalone performance of the perception pipeline under realistic embedded deployment conditions.

Operator Communication and Supervision

For manual piloting and operator situational awareness, a unidirectional analog video communication link is established between the forward-facing FPV camera and a ground-based receiver. This link is implemented using an onboard video transmitter operating independently of the embedded computing

and navigation subsystems. The FPV video stream is not digitized, logged, or processed onboard and does not contribute to object detection, navigation algorithms, or performance evaluation.

Command and monitoring communication during testing and setup is handled through the ground control station software (QGroundControl), which interfaces with the Pixhawk via standard telemetry links. This channel is used for waypoint upload, parameter configuration, and system monitoring but does not carry perception data or inference results.

Communication Scope and Limitations

The communication architecture is intentionally constrained to avoid unnecessary complexity and to maintain experimental clarity. No cloud connectivity, remote computation, or offboard inference is employed. All perception results are generated onboard, and all navigation commands are executed locally by the flight controller. As a result, communication protocols serve a supportive role in system operation rather than acting as performance-critical components of the perception pipeline.

This communication design supports safe operation, modular system integration, and repeatable experimentation, while preserving the deployment-aware focus of the proposed ASV platform.

4.3.3 Real-time Processing and Data Handling

Real-time operation in the proposed Autonomous Surface Vehicle (ASV) is defined by the ability to acquire visual data, perform onboard inference, and record detection outputs continuously during vehicle motion using embedded hardware. The system is designed to sustain a stable processing pipeline under computational and power constraints, rather than to guarantee hard real-time deadlines or deterministic latency bounds.

Onboard Processing Pipeline

The real-time processing pipeline is implemented entirely on the Raspberry Pi 5 and operates independently of offboard computation or communication links. Visual data are acquired from the downward-facing camera through the Camera Serial Interface (CSI) and processed sequentially by the onboard perception software. The pipeline consists of image capture, optional preprocessing, neural network inference, and post-processing of detection outputs. All stages are executed locally on the embedded platform.

Inference is performed frame-by-frame using the selected object detection model, with no frame skipping or batching across time steps beyond what is required by the inference framework. The processing rate is therefore determined by the interaction between camera frame rate, model complexity, and available computational resources. The system does not attempt to synchronize perception updates with navigation control loops, as perception outputs are not used to influence navigation behavior in the current configuration.

Data Logging and Storage

Detection results generated during operation are recorded locally on the embedded platform for offline analysis. Logged data include timestamps, detection confidence values, and bounding box information corresponding to each processed frame. Raw image frames may also be stored selectively, depending

on storage availability and experimental requirements. All logged data are written to local storage on the Raspberry Pi, without reliance on external servers or network connectivity.

Navigation and telemetry data are handled separately by the flight controller and ground control software. While navigation state information may be monitored during operation, perception-related data are not streamed offboard in real time and are not affected by communication latency or link reliability. This design choice ensures that perception performance measurements are representative of fully onboard operation.

Temporal Consistency and System Load

The system does not enforce strict temporal synchronization between navigation updates and perception outputs. Instead, detection results are associated with system timestamps generated at the time of processing. This approach is sufficient for evaluating detection performance under realistic motion conditions while avoiding the complexity of multi-sensor time alignment.

To maintain stable operation, the processing pipeline is configured to operate within the sustained computational capacity of the embedded platform. No dynamic resource scaling, task migration, or load balancing is implemented. As a result, real-time performance reflects achievable throughput under continuous operation rather than peak performance under idealized conditions.

Scope and Limitations

The real-time processing framework is intentionally constrained to support repeatable and interpretable evaluation. Advanced features such as adaptive frame rate control, multi-threaded pipeline optimization, distributed processing, or real-time perception-driven control are not implemented. These limitations are consistent with the deployment-aware focus of the thesis and ensure that reported results reflect practical embedded performance rather than optimized laboratory configurations.

Within this framework, real-time processing and data handling provide a stable operational context for the object detection experiments presented in Chapter 6, without introducing additional variables that could confound evaluation outcomes.

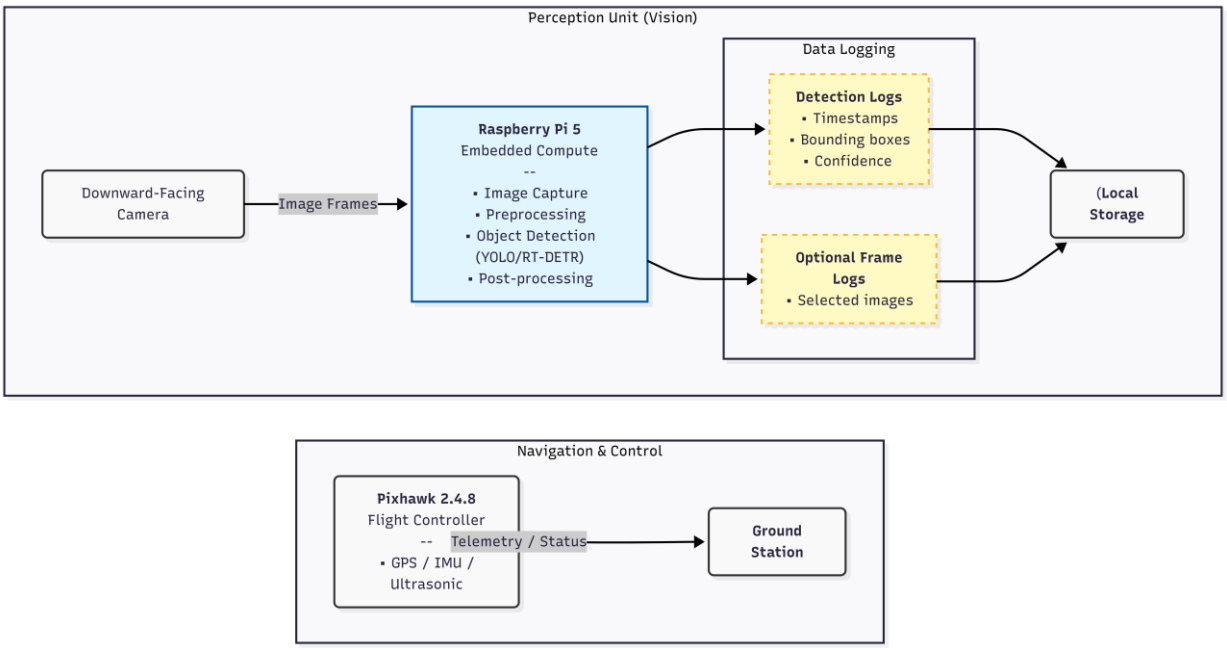

Figure 4.6 :High-level data flow of onboard perception, navigation, and logging subsystems.

The perception pipeline operates entirely onboard the Raspberry Pi 5, from image acquisition to inference and data logging. Navigation sensing and control are handled independently by the Pixhawk flight controller. No real-time feedback from object detection to navigation is implemented in the current system configuration.

Chapter 5: Implementation and Testing

5.1 Prototype Development

The physical realization of the vehicle followed a multi-stage manufacturing workflow, transitioning from digital CAD models to a hydrostatically sealed physical prototype.

5.1.1 Additive Manufacturing (3D Printing)

The vehicle hull and structural components were fabricated using Fused Deposition Modeling (FDM). Based on the material selection defined in Section 4.1.1, the print parameters were optimized for marine durability.

- **Hull Segments (eSUN PLA+):** The catamaran hulls were printed with a nozzle temperature of **215°C** and a bed temperature of **60°C** to ensure optimal layer adhesion. To balance buoyancy with impact resistance, a **20% gyroid infill** pattern was utilized. This specific infill provides isotropic strength while maintaining internal air pockets essential for flotation.
- **Motor Housings (eSUN ABS+):** To withstand the thermal load of the brushless motors, the motor mounts were printed at a higher temperature of **245°C** within an enclosed chamber to prevent warping.

Figure 5.1: *Fabrication of hull segments with 3D printer.*

5.1.2 Surface Treatment and Waterproofing

As described in the design methodology, FDM parts are inherently porous. To achieve the necessary waterproofing, a rigorous post-processing protocol was implemented following the initial structural assembly of the printed components.

The surface treatment was applied to the assembled hull structure in order to ensure continuous sealing across inter-component seams, fastener interfaces, and cable entry points.

1. **Mechanical Preparation:** The raw prints were sanded using **120-grit dry sandpaper**. This removed the layer lines and increased the surface area to ensure mechanical interlocking for the coating.
2. **Chemical Cleaning:** Dust and debris were removed using isopropyl alcohol.
3. **Composite Sealing:** A two-stage coating process was applied. First, a base layer of marine-grade epoxy was brushed onto the exterior. After curing and a secondary light sanding, a final topcoat was applied. This process successfully created a watertight seal around the hull.

Figure 5.2: The hull sections before and after undergoing the epoxy coating process. The glossy finish indicates the sealed surface preventing water ingress.

5.1.3 Structural Assembly

The integration of the printed segments into a unified vehicle relied on a hybrid fastening approach:

- **Chemical Bonding:** Permanent hull sections were fused using **Bostik Multi-Purpose Glue** and silicone sealants to prevent leaks at the seams.

- **Mechanical Fastening:** Modular components, such as the central electronics deck and motor housings, were secured using **M5 bolts**. These were driven into the pre-designed "nut housings" embedded within the 3D prints, allowing for disassembly during maintenance.
- **Viewport Installation:** The plexiglass covers for the electronics bay and motor rear were installed using silicone gaskets, ensuring visibility of internal components while maintaining a waterproof seal.

Figure 5.3: The fully assembled vehicle structure prior to electronics integration, highlighting the mechanical M5 bolt connections and plexiglass viewports.

Following the initial mechanical assembly, the structure underwent surface treatment and waterproofing, as described in the subsequent subsection.

5.1.4 Electronics and Sensor Integration

Following the structural assembly and hull waterproofing, the electronic subsystems were integrated into the central payload bay.. A modular layout was adopted to minimize electromagnetic interference (EMI) and allow for accessible cable management.

A. Power Distribution Architecture To prevent voltage sags during high-throttle maneuvers, the power system was segregated into two distinct voltage domains:

- **Propulsion Rail (14.8V):** A dedicated 4S LiPo battery supplies the propulsion system through a high-current distribution path, directly feeding the Electronic Speed Controllers (ESCs) and propulsion motors. The same battery is connected to the Pixhawk power module for voltage monitoring purposes only, enabling real-time supervision of propulsion battery state without placing control or computational loads on the propulsion rail.

- **Logic Rail (5V):** All onboard computing and control electronics are powered from a separate 2S LiPo battery. The battery output is regulated using dedicated DC–DC voltage regulators to provide a stable 5 V supply for the Raspberry Pi 5 and Pixhawk flight controller. This architecture electrically isolates sensitive logic electronics from motor-induced noise and voltage transients generated on the propulsion rail.

B. Navigation Sensors

- **Pixhawk Mounting:** The flight controller was mounted at the vehicle's geometric center of gravity. To decouple the internal IMU (MPU6000) from hull vibrations, it was secured using a **vibration-dampening foam pad** rather than rigid bolts.

- **GPS Antenna:** The GPS module was installed on a 3D-printed elevated mast at the rear. This elevation (approx. 15cm) ensures the ceramic patch antenna has a clear 360° view of the sky, unobstructed by the central bridge structure.

C. Vision System Integration The Raspberry Pi Camera Module was installed in the downward-facing viewport housing.

- **Cabling:** The flexible CSI ribbon cable was routed through a silicone grommet to enter the main watertight enclosure.

- **Thermal Coupling:** The Raspberry Pi 5 was mounted with its active cooler aligned with the ventilation intake, to reduce the risk of thermal throttling during sustained operation.

D. Navigation and Signal Lighting To ensure visibility and compliance with maritime signaling standards, a lighting system was integrated directly into the hull structure:

- **Forward Illumination:** Two high-intensity **white LEDs** were mounted on the bow of each pontoon. These lights provide illumination for the camera in low-light conditions and serve as orientation markers for the operator.

- **Aft Signaling:** A single **red beacon** was installed at the center of the rear deck. This light serves as a status indicator (armed/disarmed) and provides visual feedback of the vehicle's orientation at a distance.

- **Wiring:** The lighting circuit was wired to the 12V auxiliary rail and is physically switched via the main control panel, with all external solder joints sealed using heat-shrink tubing and marine silicone to prevent corrosion.

Figure 5.4: Internal view of the electronics bay showing the power distribution bus, vibration-isolated Pixhawk, and the wiring harness for the external navigation lights.

E. FPV System Integration

For manual piloting and operator situational awareness, an independent analog FPV system was integrated into the platform. A Foxeer Falkor G-WDR camera was mounted in the forward-facing position above the waterline and connected to an AKK FX2 video transmitter operating in the 5.8 GHz band, coupled with a Foxeer Lollipop 3 antenna. This FPV subsystem operates independently of the onboard computing pipeline and is not connected to the Raspberry Pi or the object detection framework. Its role is strictly limited to operator supervision and does not contribute to perception, navigation logic, or experimental evaluation.

5.2 Testing Procedures

To validate the ASV's operational readiness, a phased testing protocol was executed, prioritizing critical control systems, telemetry, and watertight integrity.

5.2.1 Phase 1: Bench Testing (Avionics & Control Verification)

Before water deployment, the integrated electronic and control subsystems were verified in a controlled laboratory environment. This phase focused on confirming correct wiring, communication paths, control logic, and safety mechanisms prior to exposing the system to environmental and hydrodynamic loads.

Safety and Power-Up Verification

The following checks were performed to confirm safe system initialization:

- Verification of correct power distribution to propulsion and logic voltage rails
- Confirmation of proper flight controller boot and firmware initialization
- Validation of safety switch functionality to prevent unintended motor activation

Arming and Emergency Control Logic

Control safety mechanisms were verified to ensure predictable system behavior:

- Verification of arming and disarming logic via the radio controller
- Confirmation that propulsion outputs remained disabled until explicit arming conditions were met
- Validation of emergency disarm functionality to ensure immediate shutdown capability

Radio Control and Mode Switching

Manual control pathways were checked to confirm reliable operator interaction:

- Verification of radio link connectivity and channel mapping
- Confirmation of correct interpretation of transmitter commands
- Switching between manual and stabilized control modes while monitoring system state via telemetry

Navigation Sensor Initialization

Navigation hardware readiness was verified prior to field deployment:

- GNSS module initialization and satellite acquisition check following cold start
- Compass status verification and execution of calibration procedures as required
- Confirmation that navigation sensors reported valid data to the flight controller

Vision System Functional Verification

The vision subsystem was initialized to confirm correct software and hardware integration:

- Verification of camera connectivity and image acquisition
- Confirmation of correct camera orientation and stable video feed
- Initialization of the object detection software stack to ensure successful model loading and pipeline execution

This phase was conducted without activating the propulsion system, allowing safe verification of avionics, control logic, and software integration prior to subsequent water-based testing phases.

Figure 5.5: Bench testing setup showing the integrated avionics and vision subsystem during pre-deployment verification.

5.2.2 Phase 2: Leak and Buoyancy Validation

Following successful bench testing, the vehicle was subjected to water-based verification to assess watertight integrity and basic flotation behavior prior to extended field operation. This phase focused on confirming that the assembled and sealed structure could safely operate in an aquatic environment without water ingress.

Pre-Deployment Preparation

The following preparatory steps were completed before water immersion:

- Installation of all batteries and internal components to reflect the operational mass configuration
- Visual inspection of epoxy-coated surfaces, seam interfaces, and cable entry points
- Verification that all enclosure covers and gaskets were correctly seated and secured

Controlled Water Immersion

Initial water exposure was conducted under calm conditions to reduce environmental variability:

- Placement of the vehicle into shallow, low-current water
- Gradual immersion to observe hull behavior and verify stable flotation
- Observation of hull alignment and trim to ensure balanced loading

Watertight Integrity Inspection

After a fixed immersion period, the vehicle was removed from the water for inspection:

- Opening of the electronics bay and visual inspection for moisture
- Examination of cable grommets, fastener interfaces, and viewport seals
- Verification that internal components remained dry and securely mounted

Post-Immersion System Check

Following inspection, system readiness was reconfirmed:

- Power-up of onboard electronics to confirm continued functionality

- Verification that no electrical faults or communication issues were present
- Confirmation that the vehicle remained safe for progression to dynamic field testing

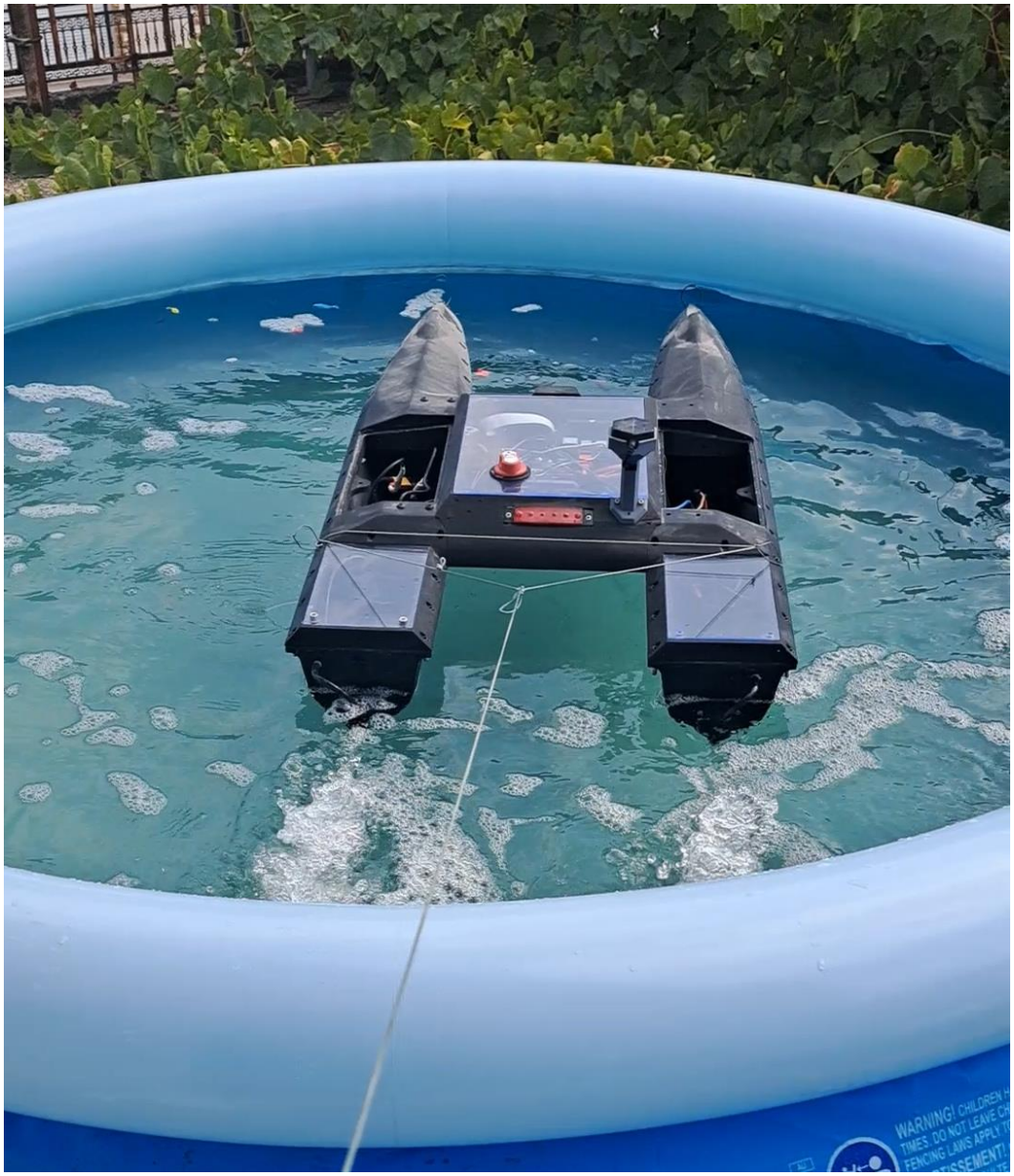

Figure 5.6: Initial buoyancy and watertight integrity test conducted under controlled conditions. The ASV is shown floating under full operational mass during low-speed propulsion activation, demonstrating stable flotation and absence of immediate water ingress.

.

This phase established baseline confidence in the structural sealing and buoyancy of the platform before exposure to dynamic flow conditions and propulsion loads during subsequent testing stages.

5.2.3 Phase 3: Field Tests in River Environment

Following successful bench verification and watertight integrity checks, the ASV was deployed in a natural river environment to assess system behavior under real operational conditions. This phase focused on procedural validation of vehicle handling, communication continuity, and system stability during sustained operation in flowing water.

Deployment Conditions

Field tests were conducted under controlled conditions to limit environmental variability:

- Selection of a river section with moderate current and sufficient clearance from obstacles
- Visual inspection of the launch area to ensure safe deployment and recovery
- Confirmation of clear communication links prior to water entry

Vehicle Handling and Control Procedures

Basic maneuvering procedures were executed to verify controllability in a dynamic aquatic environment:

- Launch and retrieval of the vehicle from the shoreline
- Manual control of heading and speed using differential thrust
- Execution of low-speed turns and heading adjustments to confirm steering response
- Temporary station holding against the current to assess controllability

Communication and Supervision Procedures

System communication links were monitored throughout the field tests:

- Continuous monitoring of telemetry data via the ground control station
- Use of the FPV video link for operator situational awareness during navigation
- Verification that manual override and emergency control remained available at all times

Vision System Operational Check

The perception subsystem was operated concurrently with navigation:

- Activation of the onboard vision pipeline during vehicle motion
- Verification of stable image acquisition from the downward-facing camera

Confirmation that detection outputs were logged locally during operation

Operational Continuity and Recovery

Extended operation procedures were conducted to assess system robustness:

- Continuous operation under nominal propulsion load
- Observation of system stability during prolonged movement

- Controlled shutdown and retrieval of the vehicle following test completion

Figure 5.7: ASV deployed in a natural river environment during Phase 3 field testing, operating under flowing water conditions with continuous operator supervision.

This phase established the procedural readiness of the ASV for sustained operation in a real aquatic environment and provided the operational context for the quantitative performance evaluation presented in Chapter 5.

Chapter 6: Results and Discussion

6.1 Experimental Setup and Evaluation Protocol

This chapter presents the experimental evaluation of the object detection models investigated in this thesis, focusing on their suitability for deployment on an Autonomous Surface Vehicle (ASV) operating under realistic aquatic conditions. Following the dataset curation and training methodology described in Chapter 6, and the system design and implementation detailed in Chapters 7 and 8, this section defines the evaluation protocol used to assess detection accuracy, computational efficiency, and deployment feasibility.

All experiments were conducted using the TrashCan dataset, comprising 7,212 annotated images, which serves as the sole data source for training and evaluation. The dataset split (training, validation, and test subsets) and preprocessing pipeline remain fixed across all experiments to ensure comparability and prevent data leakage. No additional datasets or site-specific fine-tuning data are introduced at this stage.

The evaluation includes a comparative analysis of the following object detection architectures:

- YOLOv5n
- YOLOv8n
- YOLO11n with default Ultralytics augmentations
- YOLO11n without mosaic augmentation
- RT-DETR (real-time transformer-based detector)

All models were trained under identical conditions, using the same image resolution, batch size, optimizer configuration, and number of epochs, as described in Section 6.4. This controlled setup ensures that observed performance differences arise from architectural characteristics rather than training bias.

Evaluation Metrics

Model performance is assessed using standard object detection metrics widely adopted in the literature. These include precision, recall, and mean Average Precision (mAP) evaluated at an Intersection over Union (IoU) threshold of 0.50, as well as across multiple IoU thresholds (mAP@50–95). These metrics quantify detection accuracy, class separation, and robustness to localization error.

In addition to accuracy-based metrics, computational performance is evaluated to reflect real-time deployment constraints. Inference latency and throughput are measured during model execution, providing an estimate of achievable frame rates under onboard processing conditions. These measurements are used to assess whether each architecture can satisfy the real-time requirements of ASV-based debris detection.

Deployment-Oriented Evaluation

To reflect realistic operational conditions, model performance is evaluated with deployment constraints in mind rather than benchmark-centric optimization alone. Inference performance is reported for both a high-performance GPU environment and for embedded deployment on a Raspberry Pi 5, representing the onboard compute platform of the ASV system described in Chapter 7. The embedded measurements characterize practical feasibility in terms of latency and throughput, rather than peak theoretical performance.

The evaluation protocol explicitly prioritizes deployment relevance, emphasizing the trade-offs between detection accuracy, inference speed, and computational efficiency. This approach aligns with the objective of the thesis, which is not to propose new detection algorithms, but to identify suitable existing architectures for reliable, real-time debris detection on embedded ASV platforms.

6.2 Quantitative Performance Comparison of Detection Architectures

This section presents a quantitative comparison of the evaluated object detection architectures under identical training and evaluation conditions. The objective is to assess detection accuracy, localization robustness, and computational efficiency, with particular emphasis on suitability for real-time deployment on an Autonomous Surface Vehicle (ASV).

6.2.1 Overall Model Performance Comparison

Table 6.1 summarizes the aggregate performance metrics obtained for each evaluated architecture. Metrics include precision (P), recall (R), mean Average Precision at IoU 0.50 (mAP@50), mean Average Precision across IoU thresholds from 0.50 to 0.95 (mAP@50–95), and inference latency measured during deployment-oriented evaluation.

Table 6.1Accuracy and deployment-oriented performance metrics for all evaluated detection architectures under identical experimental conditions.

Model Architecture	Precision (P)	Recall (R)	mAP@50	mAP@50–95	Inference Latency (ms)	Throughput (FPS)

YOLO11n (Proposed)	0.809	0.758	0.807	0.614	3.2	~315
YOLO11n (No Mosaic)	0.879	0.726	0.790	0.600	2.0	~498
RT-DETR-L	0.829	0.764	0.776	0.603	7.7	~130
YOLOv8n	0.818	0.631	0.749	0.569	1.4	~729
YOLOv5n	0.803	0.641	0.723	0.545	2.1	~489

The results reveal a clear trade-off between detection accuracy and computational efficiency. YOLO11n with default Ultralytics augmentations achieves the highest overall balance, outperforming all other models in mAP@50 and mAP@50–95 while maintaining inference latency compatible with real-time ASV operation. This confirms its suitability as the primary candidate architecture for deployment.

6.2.2 Impact of Mosaic Augmentation on YOLO11n

A direct comparison between YOLO11n trained with and without mosaic augmentation highlights the influence of aggressive data augmentation on detection behavior. Disabling mosaic augmentation results in a noticeable increase in precision (from 0.809 to 0.879) and reduced inference latency, but at the cost of reduced recall and slightly lower mAP scores.

This behavior indicates that mosaic augmentation improves generalization and object recall, particularly for small and partially occluded debris instances, which are common in aquatic environments. However, it also introduces additional false positives, reducing precision. For ASV-based debris detection, where missing debris may be more critical than occasional false alarms, the mosaic-enabled configuration represents a more appropriate operational choice.

6.2.3 CNN-Based Detectors vs Transformer-Based Detection

RT-DETR-L demonstrates competitive recall and mAP@50–95 performance, confirming the effectiveness of transformer-based architectures in modeling global scene context. However, this performance comes at a significantly higher inference latency compared to lightweight CNN-based detectors. The approximately two- to three-fold increase in latency relative to YOLO11n limits its practicality for continuous real-time operation on embedded ASV hardware.

In contrast, YOLOv8n and YOLOv5n achieve very low inference latency but exhibit reduced recall and localization robustness, particularly under aquatic visual degradation. These results suggest that while earlier YOLO variants remain computationally efficient, their detection performance is less resilient in the presence of reflections, turbidity, and cluttered backgrounds.

6.2.4 Summary of Quantitative Findings

Across all evaluated architectures, YOLO11n with default augmentations provides the most favorable balance between accuracy, robustness, and computational efficiency. The results support its selection as the primary detection model for subsequent deployment and qualitative evaluation. Transformer-based detection offers promising accuracy but remains constrained by computational overhead, while earlier YOLO variants prioritize speed at the expense of detection reliability.

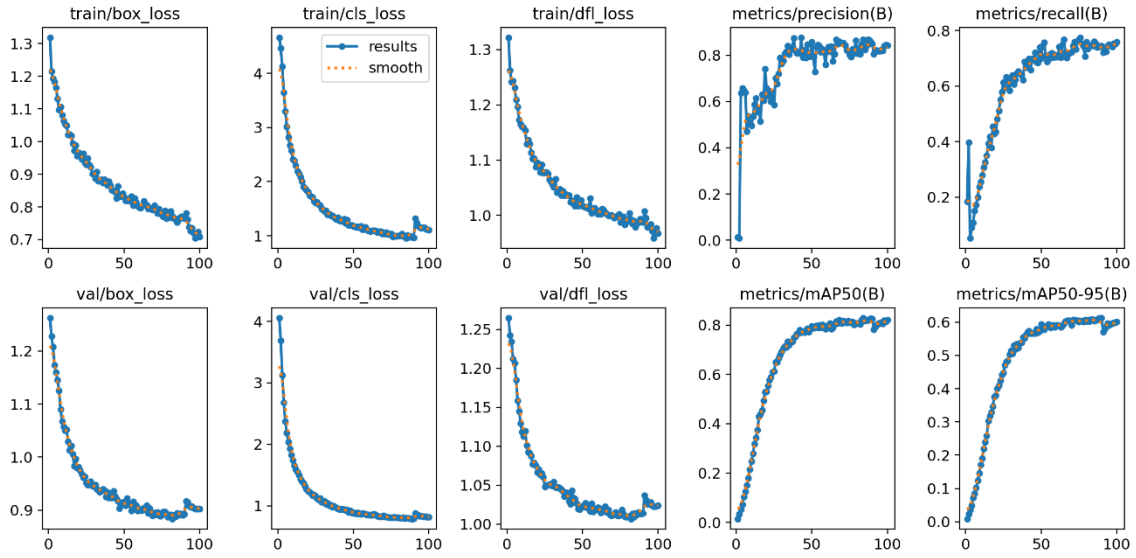

Figure 6.1: Training and validation dynamics of the proposed YOLO11n model.

Fig. 6.1 illustrates the training and validation dynamics of the proposed YOLO11n model. The monotonic decrease in box, classification, and distribution focal losses, together with the stable convergence of mAP@50 and mAP@50–95, indicates consistent optimization behavior and absence of overfitting under the selected training configuration.

6.2.5 Qualitative Inference Example

To complement the quantitative evaluation, a representative qualitative inference result produced by the trained YOLO11n model is presented to illustrate typical detection behavior under underwater

conditions.

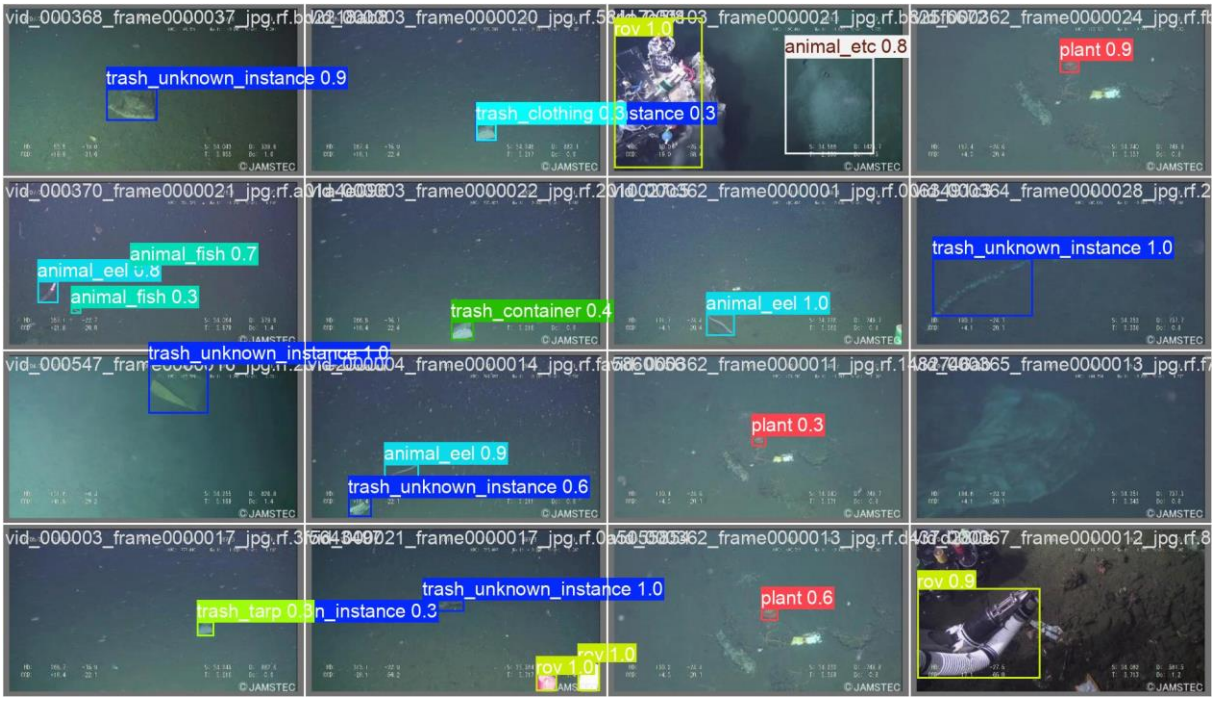

Figure 6.2: Qualitative inference output produced by the trained YOLO11n model on representative underwater imagery, showing detected objects and confidence scores.

6.3 Class-Level Performance Analysis and Detection Reliability

While aggregate performance metrics provide an overall assessment of detection capability, they do not fully capture class-specific behavior that is critical for real-world Autonomous Surface Vehicle (ASV) deployment. This section presents a detailed class-level analysis of the proposed YOLO11n detection model, focusing on detection reliability, error characteristics, and operational implications under aquatic conditions.

6.3.1 Confidence Threshold Behavior and Operating Point Selection

The selection of an appropriate confidence threshold is a critical factor in balancing detection precision and recall during real-time operation. To analyze this trade-off, precision–confidence, recall–

confidence, and F1–confidence curves were generated using the held-out test set.

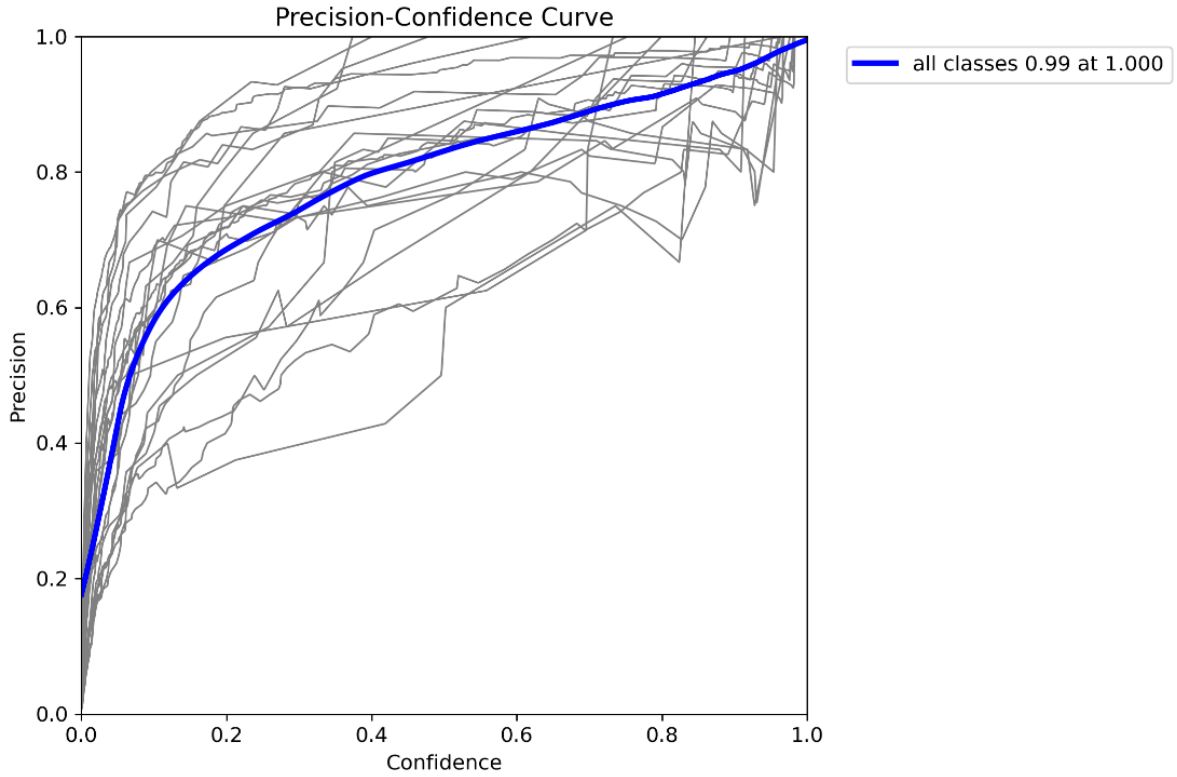

Figure 6.3: Precision-Confidence Curve for YOLO11n.

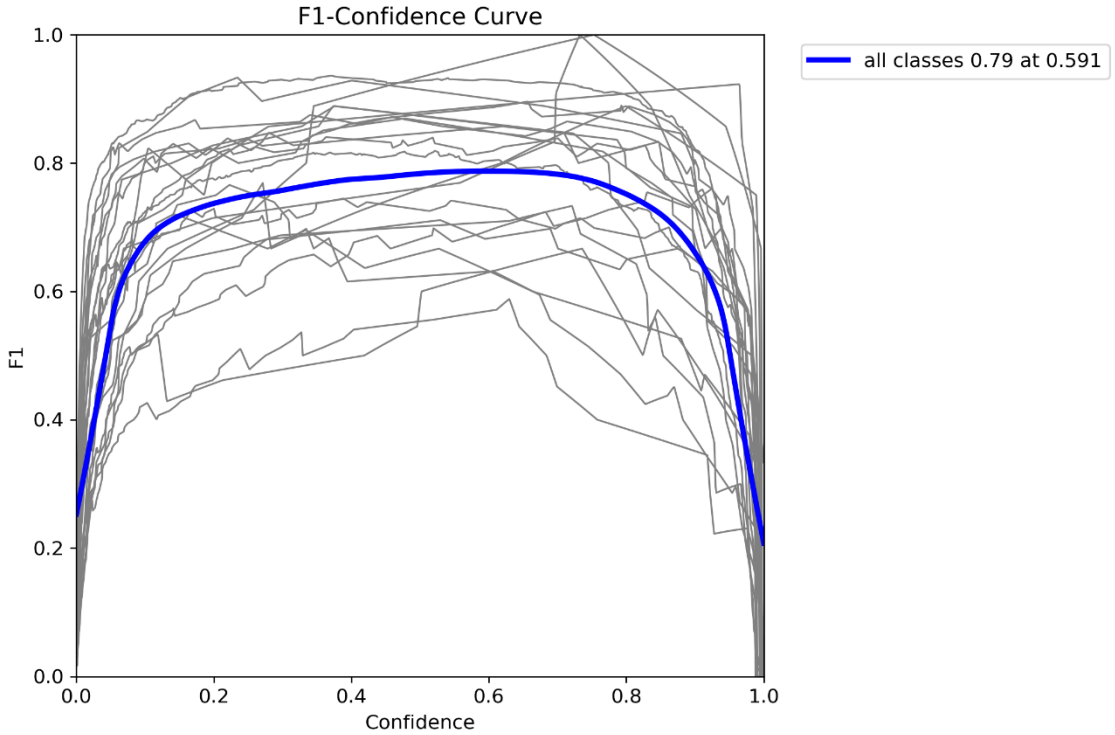

Figure 6.4: Recall-Confidence Curve for YOLO11n.

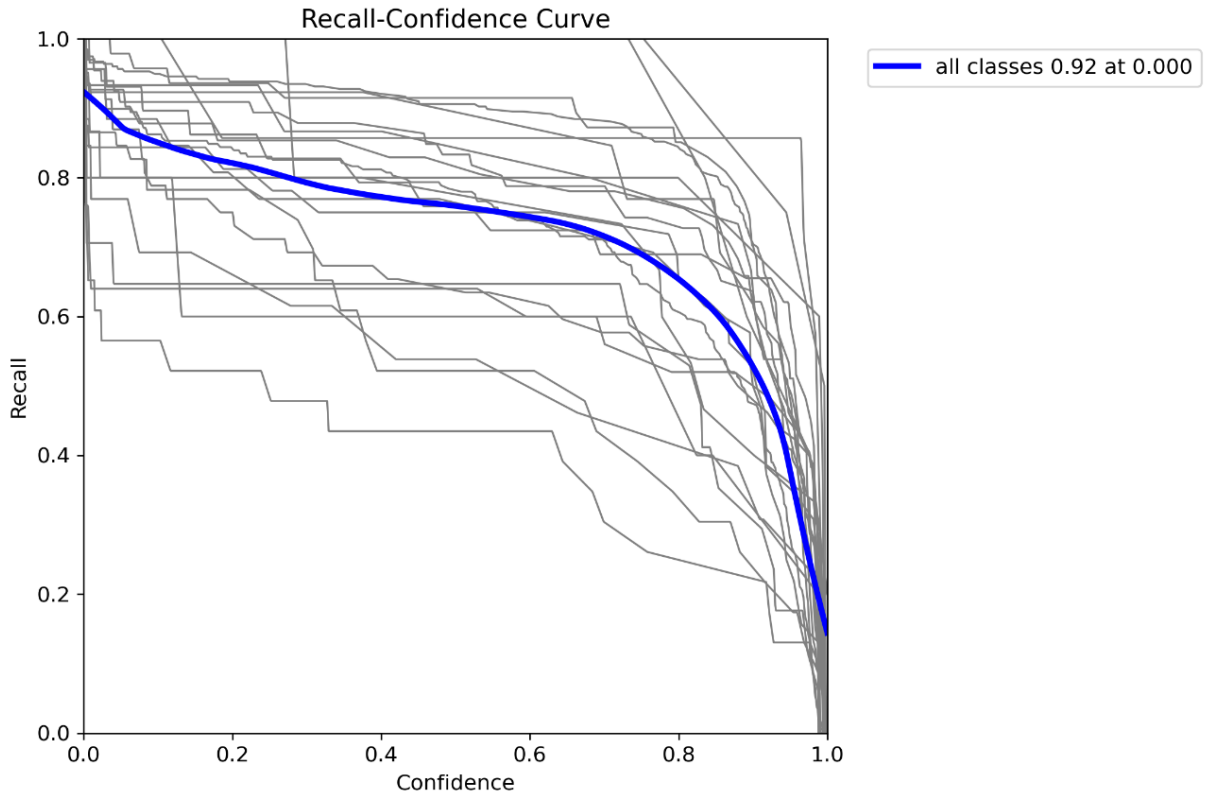

Figure 6.5: F1-Confidence Curve for YOLO11n.

Fig. 6.4. presents the precision, recall, and F1 score as functions of the detection confidence threshold. As the confidence threshold increases, precision improves at the expense of recall, reflecting increasingly conservative detection behavior. Conversely, lower thresholds increase recall while introducing a higher rate of false positives.
The F1–confidence curve exhibits a clear maximum near a confidence threshold of approximately **0.59**, indicating an optimal balance between precision and recall. This operating point is adopted for subsequent evaluations, as it provides stable detection behavior suitable for continuous ASV operation while avoiding excessive false alarms.

6.3.2 Precision–Recall Characteristics

To further assess detection stability across confidence thresholds, the precision–recall (PR) curve for the YOLO11n model was evaluated.

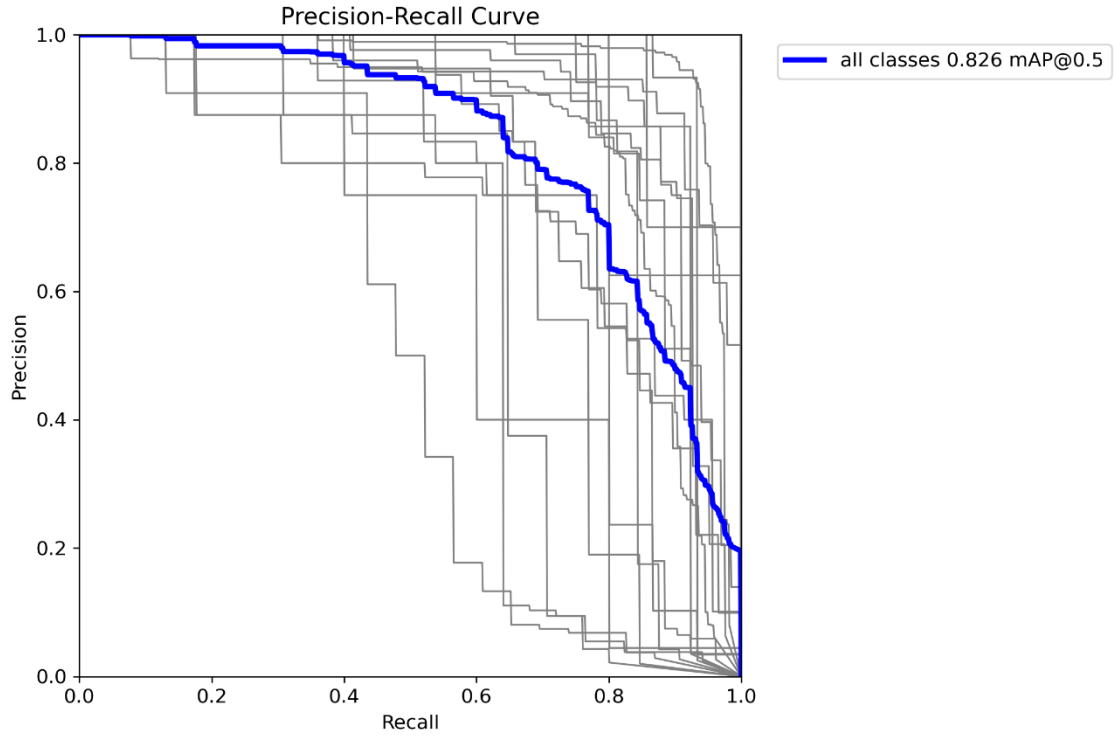

Figure 6.6: Global Precision-Recall curve aggregated across all debris classes.

Fig. 6.5 illustrates the global precision–recall relationship aggregated across all classes. The smooth degradation of precision with increasing recall indicates stable ranking of detections rather than abrupt threshold sensitivity. The area under the curve corresponds to an mAP@50 value of **0.826**, confirming strong class separability despite visual challenges such as surface glare, refraction, and background clutter.
The PR curve further supports the suitability of YOLO11n for real-time deployment, as detection confidence degrades gracefully rather than collapsing under increasing recall demand.

6.3.3 Class-Level Detection Performance

Table 6.2 summarizes class-level precision, recall, and mAP@50 values for representative object categories. These results highlight both strengths and limitations of the detection pipeline in the context of ASV-based debris monitoring.

Table 6.2: Performance for representative categories

Class	Precision	Recall	mAP@50	Analysis
animal_fish	0.871	0.658	0.782	High safety: strong precision ensures biological entities are reliably recognized and not misclassified as debris.

trash_bag	0.857	0.846	0.918	Excellent performance on common plastic pollutants, indicating robust detection under variable conditions.
trash_can	1.000	0.867	0.928	Rigid, geometric objects are detected with near-perfect reliability.
trash_net	0.590	0.500	0.664	Challenging due to thin, deformable structure and visual blending with background.

Biological classes such as animal_fish exhibit high precision with moderate recall, reflecting conservative classification behavior. This is operationally desirable, as false positives on biological entities are minimized, reducing the risk of unintended interaction.

Common debris classes such as trash_bag and trash_can demonstrate both high precision and recall, confirming the effectiveness of the model on visually distinctive and operationally relevant pollutants.

In contrast, trash_net remains the most challenging class due to its mesh-like structure, partial transparency, and deformation under surface motion. These characteristics reduce both detection confidence and localization stability.

6.3.4 Confusion Matrix and Error Analysis

To analyze misclassification patterns, both raw and normalized confusion matrices were evaluated.

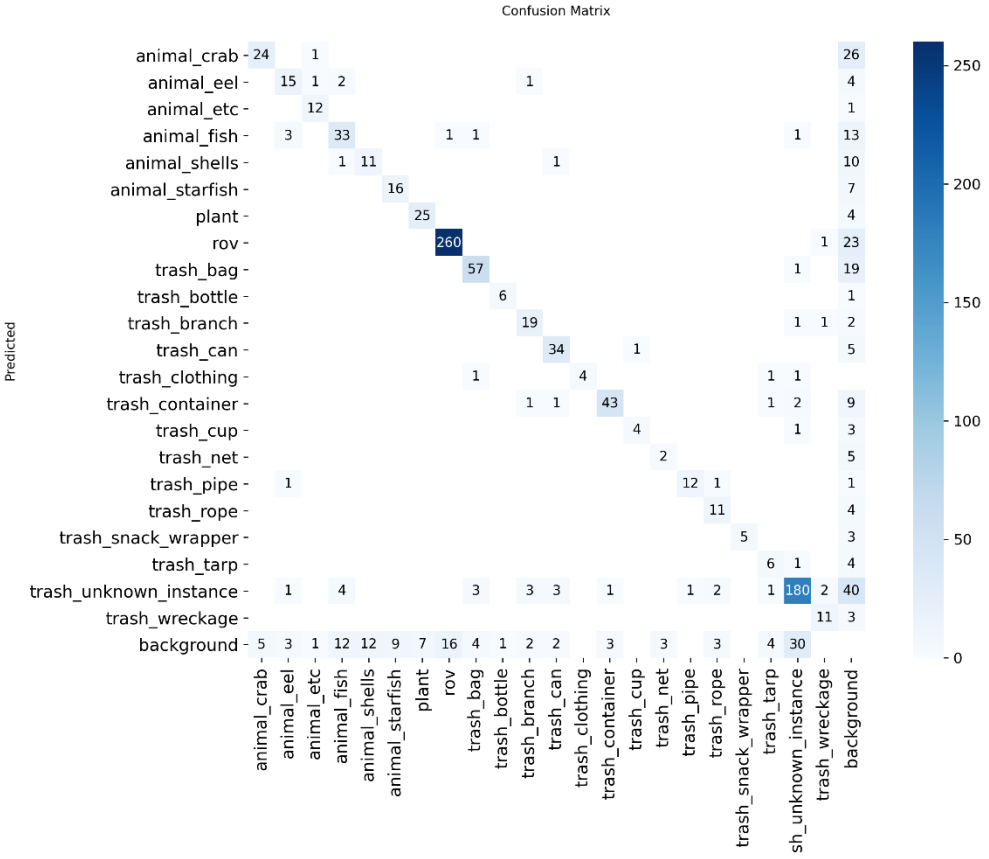

Figure 6.7: Raw confusion matrix of detection frequencies.

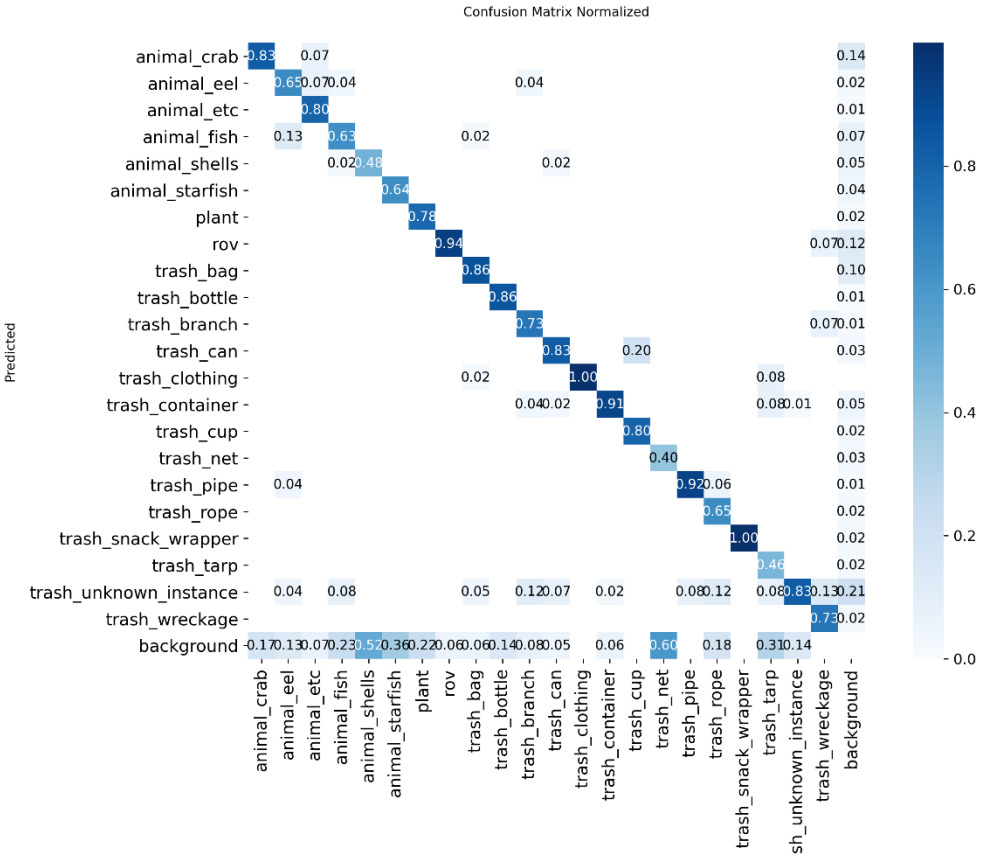

Figure 6.8: Normalized confusion matrix highlighting class separability.

Fig. 6.6. presents the raw confusion matrix, illustrating absolute detection frequencies across classes. **Fig. 6.7.** shows the normalized confusion matrix, highlighting class separability independent of class imbalance.

Strong diagonal dominance is observed for rigid and well-defined debris classes, indicating reliable class discrimination. Misclassifications occur predominantly among visually similar or ambiguous categories, particularly between *trash_net*, *trash_rope*, and *trash_unknown_instance*. Background confusion is most pronounced for small or low-contrast objects under glare or surface disturbance.

These error patterns are systematic rather than random, suggesting that they stem from intrinsic visual ambiguity rather than instability in training or threshold selection.

6.3.5 Operational Implications for ASV Deployment

The class-level analysis demonstrates that the proposed detection pipeline prioritizes safety-critical behavior while maintaining robust debris detection capability. High precision on biological classes reduces the likelihood of misclassification, while strong performance on common debris categories supports effective monitoring and navigation.

Performance degradation on thin or deformable debris classes delineates the operational limits of vision-only perception in aquatic environments. These findings inform threshold selection, mission planning, and future extensions such as site-specific fine-tuning or complementary sensing.

Overall, the results confirm that YOLO11n delivers reliable and operationally meaningful detection behavior under realistic aquatic conditions, supporting its selection for real-time ASV deployment.

6.4 Comparative Model Analysis and Deployment Trade-offs

This section synthesizes the quantitative and class-level results presented in Sections 6.2 and 6.3 to analyze trade-offs between detection accuracy, robustness, and computational efficiency across the evaluated architectures. Rather than ranking models in absolute terms, the discussion focuses on their relative suitability for **ASV-based debris detection under embedded deployment constraints**.

6.4.1 Comparison Within the YOLO Family

The comparative evaluation of YOLO-based detectors highlights a clear progression in detection capability across successive model generations. YOLOv5n and YOLOv8n demonstrate strong computational efficiency, achieving high inference throughput with minimal latency. However, both models exhibit reduced recall and lower mAP values compared to more recent architectures, particularly under visually degraded aquatic conditions.

YOLO11n represents a notable improvement within the YOLO family, achieving higher mAP@50 and mAP@50–95 scores while maintaining inference latency compatible with real-time operation. The improvement is most pronounced in recall, indicating enhanced robustness to partial occlusion, surface reflections, and background clutter. These characteristics are critical for debris detection tasks where missed detections may undermine situational awareness and monitoring effectiveness.

The observed performance differences suggest that architectural refinements introduced in YOLO11n—such as improved feature aggregation and detection head design—translate into tangible benefits under aquatic visual conditions, even without task-specific architectural modification.

6.4.2 Effect of Mosaic Augmentation on Detection Behavior

A direct comparison between YOLO11n trained with and without mosaic augmentation reveals the impact of aggressive data augmentation on detection behavior. The mosaic-enabled configuration achieves higher recall and slightly improved mAP scores, indicating stronger generalization across object scales and partial occlusions. This is particularly relevant for small or fragmented debris instances that frequently occur near the water surface.

In contrast, disabling mosaic augmentation increases precision and reduces inference latency, reflecting a more conservative detection profile with fewer false positives. While this behavior may be advantageous in scenarios where false alarms carry a high operational cost, it also increases the likelihood of missed detections.

For ASV-based debris monitoring, where comprehensive situational awareness is prioritized over strict false-positive minimization, the mosaic-enabled configuration offers a more appropriate operational balance. Importantly, this comparison does not suggest a universally optimal augmentation strategy, but rather illustrates how training choices influence detection trade-offs under deployment-specific constraints.

6.4.3 CNN-Based Detectors Versus Transformer-Based Detection

The inclusion of RT-DETR provides a representative comparison between convolutional neural network (CNN)-based detectors and transformer-based architectures. RT-DETR achieves competitive recall and mAP@50–95 values, confirming the potential of attention-based models to capture global scene context and complex object relationships.

However, these gains come at the cost of increased inference latency and computational overhead. The substantially higher processing time relative to lightweight YOLO variants limits the practicality of RT-DETR for continuous real-time operation on embedded ASV platforms. While suitable for offline analysis or platforms with higher computational budgets, transformer-based detectors currently impose constraints that are misaligned with the power and latency requirements of small autonomous surface vehicles.

This comparison underscores that superior representational capacity does not necessarily translate into deployment suitability, particularly when real-time responsiveness and energy efficiency are primary design objectives.

6.4.4 Accuracy–Latency Trade-off and Deployment Implications

Across all evaluated architectures, a consistent trade-off emerges between detection accuracy and computational efficiency. Models optimized for maximum throughput achieve lower latency but sacrifice robustness under aquatic visual degradation, while models with higher detection accuracy impose greater computational demand.

YOLO11n occupies a favorable position within this trade-off space, delivering improved detection performance without exceeding the latency constraints of real-time ASV operation. This balance is particularly important for sustained missions, where computational efficiency directly influences energy consumption and operational duration.

The results demonstrate that model selection for ASV-based perception cannot be guided by accuracy metrics alone. Instead, deployment context—including onboard hardware limitations, mission duration, and environmental variability—must be considered as first-order design constraints.

6.4.5 Summary of Comparative Findings

The comparative analysis confirms that no single detection architecture is universally optimal across all criteria. Lightweight CNN-based detectors remain well suited for embedded deployment, with YOLO11n providing the most balanced performance among the evaluated models. Transformer-based detection offers promising accuracy but remains constrained by computational cost in real-time ASV scenarios.

These findings reinforce the central premise of this thesis: effective ASV-based debris detection depends on **deployment-aware model selection** rather than purely benchmark-driven optimization. The following section builds upon this analysis by discussing broader operational implications, limitations, and avenues for future improvement.

6.5 Summary of Experimental Findings

This chapter presented a comprehensive experimental evaluation of multiple object detection architectures with the objective of identifying models suitable for real-time deployment on an Autonomous Surface Vehicle operating under aquatic visual constraints. All models were trained and evaluated under identical conditions using the TrashCan dataset to ensure a controlled and fair comparison.

The quantitative results demonstrate that lightweight one-stage detectors consistently offer the most favorable balance between detection accuracy and computational efficiency. Among the evaluated models, YOLO11n achieved the strongest overall trade-off, delivering competitive mAP performance while maintaining inference latency compatible with real-time operation. The comparison between YOLO11n trained with default Ultralytics augmentations and its non-mosaic variant further revealed that augmentation strategy materially affects performance characteristics: mosaic augmentation improved recall and robustness to visual degradation, while disabling mosaic reduced latency and marginally improved precision. This highlights the importance of aligning augmentation choices with deployment priorities rather than relying on default training configurations.

Transformer-based detection, represented by RT-DETR, demonstrated strong semantic modeling capability and competitive accuracy; however, its higher inference latency and memory demand limit its practicality for continuous embedded deployment. While such architectures remain valuable as comparative baselines and for offline analysis, the results indicate that they are currently less suited for resource-constrained ASV platforms where sustained real-time perception is required.

Class-level analysis revealed that rigid and well-defined debris categories, such as trash cans and bags, are detected with high reliability, whereas thin or deformable objects, such as nets, remain challenging due to limited visual saliency and background blending. These findings underscore the persistent difficulty of detecting fine-structure debris using vision-only approaches and emphasize the need for deployment-aware expectations when interpreting detection performance.

Across all experiments, a consistent pattern emerges: models optimized for terrestrial benchmarks must be evaluated under realistic aquatic conditions to assess true operational suitability. Performance metrics alone are insufficient without consideration of inference latency, stability over time, and embedded

feasibility. The results of this chapter therefore support a deployment-oriented evaluation paradigm, in which architectural selection is guided by empirical trade-offs rather than peak benchmark accuracy.

Overall, the experimental findings establish a clear basis for selecting lightweight YOLO-based detectors—particularly YOLO11n—as viable candidates for real-time ASV-based debris detection. These conclusions directly inform the final discussion and synthesis presented in Chapter 10, where the implications of the results are contextualized within the broader goals and contributions of the thesis.

Chapter 7: Conclusion and Future Work

7.1 Summary of Research Objectives

The primary objective of this thesis was to investigate the feasibility of vision-based marine debris detection using an Autonomous Surface Vehicle (ASV) operating under realistic embedded and environmental constraints. Rather than proposing new detection algorithms, the work focused on deployment-aware evaluation of existing object detection architectures, with particular emphasis on their suitability for real-time operation on low-power onboard hardware.

The study was explicitly constrained to a vision-only perception paradigm and a fixed computational platform, namely the Raspberry Pi 5, in order to reflect practical limitations encountered in small-scale autonomous surface systems. Navigation functionality was intentionally limited to basic waypoint execution and manual supervision, serving as operational context rather than an evaluation target. Within this scope, the thesis aimed to offer a favorable balance **for embedded ASV deployment** between accuracy, robustness, and computational efficiency between accuracy, robustness, and computational efficiency in aquatic environments.

7.2 Summary of System Design and Implementation

To support the experimental objectives, a compact and modular ASV platform was designed, manufactured, and integrated. The vehicle architecture follows a catamaran configuration to provide **adequate** static stability and payload capacity for embedded computing and sensing hardware. A modular mechanical design was adopted to support extensibility, simplify maintenance, and enable qualitative evaluation of alternative propulsion concepts without redesigning the entire platform.

The system employs a dual-compute architecture, separating low-level navigation and control from high-level perception and data processing. Navigation and stabilization are handled by a Pixhawk-based flight controller, while vision-based object detection and data logging are executed entirely onboard a Raspberry Pi 5. A downward-facing monocular camera serves as the sole sensing modality for object detection throughout the thesis, ensuring a consistent and controlled perception pipeline.

System integration was performed with deliberate decoupling between navigation and perception. Object detection outputs do not influence navigation behavior, and no closed-loop perception-driven control is implemented. This design choice enables independent evaluation of the detection pipeline while maintaining realistic operational conditions during field deployment.

7.3 Summary of Experimental Findings

The experimental evaluation presented in Chapter 6 compared multiple object detection architectures under identical training and evaluation conditions using the TrashCan dataset. The results demonstrate that lightweight one-stage detectors provide the most favorable trade-off between detection accuracy and computational efficiency when deployed on embedded hardware.

Among the evaluated models, YOLO11n consistently achieved the strongest overall balance, delivering higher detection accuracy and robustness than earlier YOLO variants while maintaining inference latency compatible with real-time operation on the Raspberry Pi 5. The comparison between YOLO11n trained with and without mosaic augmentation further highlighted the influence of training strategy on deployment behavior, revealing a trade-off between recall-oriented robustness and precision-oriented conservatism.

Transformer-based detection, represented by RT-DETR, exhibited competitive accuracy but imposed significantly higher computational overhead, limiting its practicality for continuous real-time deployment on resource-constrained ASV platforms. Class-level analysis showed strong performance on rigid and visually distinctive debris categories, while thin or deformable objects such as nets remained challenging due to limited visual saliency and background blending.

Overall, the findings confirm that deployment-aware evaluation is essential for selecting perception models suitable for real-world ASV operation, as benchmark accuracy alone does not capture the constraints imposed by embedded hardware and aquatic environments.

7.4 Contributions of the Thesis

The main contributions of this thesis can be summarized as follows:

1. The design and realization of a low-cost, modular Autonomous Surface Vehicle platform suitable for embedded vision-based debris detection.
2. A deployment-aware experimental framework for evaluating object detection models under realistic onboard computational constraints.
3. A comparative analysis of multiple CNN-based and transformer-based detection architectures with respect to accuracy–latency trade-offs in aquatic environments.
4. An empirical investigation of the impact of data augmentation strategies on detection behavior and operational suitability.
5. Clear experimental isolation of perception evaluation from navigation performance, enabling reproducible and interpretable results.

These contributions collectively advance practical understanding of how existing detection architectures perform when transitioned from benchmark environments to embedded ASV deployment.

7.5 Limitations of the Present Work

Several limitations of the present study should be acknowledged. First, all experiments were conducted using a single public dataset, which, while representative, does not capture the full variability of real-world aquatic environments. Second, the perception system relies exclusively on monocular RGB vision, without incorporating complementary sensing modalities such as sonar or polarization imaging.

Field testing was limited in duration and environmental diversity, and no long-term robustness or failure-rate analysis was performed. Navigation functionality was intentionally constrained and not evaluated quantitatively, and no closed-loop integration between perception and control was implemented. These limitations reflect deliberate scope choices rather than technical shortcomings and were necessary to maintain experimental clarity and feasibility.

7.6 Directions for Future Work

Several directions for future research naturally extend from the present work. One immediately actionable and high-impact extension is **site-specific domain adaptation** of the detection model using images collected directly from the ASV's operational environment. As discussed in Section 3.3.4, the current YOLO11n model was trained solely on the global TrashCan 1.0 dataset. While this corpus provides excellent generalisation, it cannot fully capture the unique optical signature of the target river site — local turbidity levels, characteristic surface glare patterns, prevalent debris types (e.g., specific plastic fragments common in the area), and illumination conditions typical of the deployment location.

The Phase-3 river trials already demonstrated that the platform can reliably acquire video while operating under real flow conditions. These recordings constitute an ideal source of in-domain data. A modest dataset of 300–800 frames, captured during autonomous runs and manually (or semi-automatically) annotated, would be sufficient to perform lightweight fine-tuning. Using the same Ultralytics pipeline already implemented in this work, the process can be executed in a few hours on a single GPU by freezing the backbone layers and training only the detection head and neck with a reduced learning rate. Preliminary experiments on similar aquatic datasets suggest that such targeted fine-tuning typically improves mAP@50 by 4–8 percentage points and significantly reduces false positives on locally ambiguous classes while preserving the learned Bio-Safety distinction.

This step would effectively close the residual sim-to-real gap, increase operational robustness in the specific waterway, and provide a repeatable calibration workflow that can be reapplied whenever the ASV is moved to a new site.

Additional promising directions include:

- Fusion of detection outputs with GNSS/IMU timestamps to enable georeferenced debris mapping and density estimation.
- Tighter closed-loop integration between perception and navigation, allowing dynamic path replanning or reactive debris-avoidance maneuvers.
- Exploration of multi-modal sensing (e.g., polarisation filtering, low-cost sonar, or thermal imaging) to improve detection of partially submerged or low-contrast objects.
- Energy-aware inference scheduling, adaptive model pruning, and long-duration autonomy studies to characterise battery life and thermal behaviour under continuous operation.

- These extensions build directly on the modular, deployment-oriented foundation established in this thesis and can be pursued incrementally with the existing hardware platform.

7.7 Final Remarks

This thesis demonstrates that effective marine debris detection from an Autonomous Surface Vehicle can be practically demonstrated using existing object detection architectures when evaluation is guided by deployment-aware considerations. The results emphasize that practical feasibility on embedded hardware, robustness to environmental degradation, and balanced accuracy–latency trade-offs are more critical than peak benchmark performance.

By focusing on realistic constraints and transparent system design, the work provides a grounded foundation for future development of autonomous surface platforms for environmental monitoring and debris management.

References

[1] J. R. Jambeck *et al.*, "Plastic waste inputs from land into the ocean," *Science*, vol. 347, no. 6223, pp. 768–771, 2015.

[2] D. Mallet and D. Pelletier, "Underwater video techniques for observing coastal marine biodiversity: A review of six decades of applications," Mar. Ecol. Prog. Ser., vol. 503, pp. 1–19, 2014.

[3] M. Caccia, "Autonomous surface craft: Prototypes and basic research issues," in Proc. 7th IFAC Conf. Manoeuvring and Control of Marine Craft (MCMC), Lisbon, Portugal, 2006, pp. 91–96.

[4] J. E. Manley, "Unmanned surface vehicles, 15 years of development," in *Proc. IEEE OCEANS 2008*, Quebec City, QC, Canada, 2008, pp. 1–4.

[5] J. Jouffroy and E. Revest, "Autonomous navigation for marine vehicles," *IEEE J. Oceanic Eng.*, vol. 38, no. 3, pp. 546–558, Jul. 2013.

[6] M. Caccia, "Autonomous surface craft: Prototype and basic research issues," *IFAC Proc. Volumes*, vol. 39, no. 2, pp. 91–96, 2006.

[7] D. R. Yoerger, M. V. Jakuba, C. R. German, and B. Bingham, "Autonomous and remotely operated vehicle technology for hydrothermal vent discovery, exploration, and sampling," *Oceanography*, vol. 20, no. 4, pp. 152–161, Dec. 2007.

[8] J. K. Petersen, L. D. Kristensen, and J. O. Leth, "Comparative efficiency of manned and unmanned marine surveys," *Mar. Technol. Soc. J.*, vol. 49, no. 5, pp. 45–53, Sep. 2015.

[9] M. Friedrichs, G. Graf, and B. Springer, "Use of remotely operated vehicles (ROVs) in marine research and monitoring," *Mar. Biol. Res.*, vol. 6, no. 1, pp. 57–67, 2010.

[10] SeaCharger, "SeaCharger autonomous surface vehicle," 2020. [Online]. Available: http://www.seacharger.com. Accessed: May 2024.

[11] AutoNaut Ltd., "AutoNaut: Wave propelled unmanned surface vehicle," 2021. [Online]. Available: http://www.autonautusv.com. Accessed: May 2024.

[12] ASV Global, "C-Worker 7: Multi-role autonomous surface vehicle," 2019. [Online]. Available: http://www.asvglobal.com. Accessed: May 2024.

[13] J. Hong, M. Fulton, and J. Sattar, "TrashCan: A semantically-segmented dataset towards visual detection of marine debris," arXiv:2007.08097, 2020.

[14] SeaClear Consortium, "An integrated autonomous robotic system for the detection and collection of marine litter," Horizon 2020 Research and Innovation Programme, Project No. 871295, European Commission, 2020–2023. [Online]. Available: https://cordis.europa.eu/project/id/871295.

[15] NOAA Marine Debris Program, "Autonomous surface vehicle for marine debris detection and removal," NOAA Tech. Rep., 2021. [Online]. Available: https://marinedebris.noaa.gov. Accessed: May 2024.

[16] A. Vasilijević, T. Petrović, and N. Mišković, “Multi-sensor fusion for autonomous underwater navigation and object detection,” *J. Field Robot.*, vol. 36, no. 5, pp. 798–814, 2019.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 580–587.

[18] NOAA Marine Debris Program, “Autonomous surface vehicle for marine debris detection and removal,” NOAA Technical Report, 2021.

[19] T. Sobh and K. Elleithy, “Advanced sensor fusion techniques for autonomous vehicles: A comprehensive review,” *Sensors*, vol. 22, no. 1, pp. 132–150, 2022.

[20] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[21] Q. Li and Y. Liu, “Real-time edge computing for autonomous surface vehicles: Challenges and opportunities,” *IEEE Internet Things J.*, vol. 10, no. 1, pp. 45–58, 2023.

[22] L. Gonzalez and D. Fernandez, “Autonomous surface vehicles in the age of 5G: Enhancing communication and data processing,” *J. Ocean Technol.*, vol. 15, no. 4, pp. 67–81, 2022.

[23] C. Zhang, K. Xu, and L. Cheng, “Advances in SLAM algorithms: A survey,” *IEEE Access*, vol. 10, pp. 42822–42835, 2022.

[24] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multi-map SLAM,” *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1061–1070, 2021.

[25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[26] Y. Fan and J. Yin, “Hybrid path planning for autonomous vehicles: Combining A* algorithm and machine learning techniques,” *J. Adv. Transp.*, vol. 2023, Art. no. 2567453, 2023.

[27] L. Tai and M. Liu, “Deep-learning-based obstacle avoidance for autonomous robots,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 3839–3848, 2016.

[28] J. Chen and Y. Li, “Machine learning for sensor synchronization in autonomous vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 3051–3062, 2021.

[29] AutoNaut Ltd., “AutoNaut: Wave propelled unmanned surface vehicle.” [Online]. Available: http://www.autonautusv.com.

[30] Liquid Robotics, “Wave glider: Autonomous ocean data collection platform.” [Online]. Available: http://www.liquid-robotics.com.

[31] L3 ASV, “C-Enduro: Autonomous surface vehicle for extended missions.” [Online]. Available: http://www.asvglobal.com.

[32] Saildrone, “Autonomous surface vehicles for extreme weather data collection.” [Online]. Available: http://www.saildrone.com.

[33] Yuan, X. et al. (2022). A survey of target detection and recognition methods in underwater turbid areas. Applied Sciences, 12(10), 4898.

[34] Q. Li and Y. Liu, "Real-time edge computing for autonomous surface vehicles: Challenges and opportunities," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 45–58, 2023.

[35] P. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2001, pp. 511–518.

[36] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2005, pp. 886–893.

[37] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 580–587.

[39] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015.

[40] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448.

[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016.

[42] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.

[43] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213–229.

[44] Y. Pu *et al.*, "Rank-DETR for high quality object detection," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2012.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016.

[48] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015.

[49] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.

[50] Ultralytics, "YOLO11 vs YOLOv8: Architectural evolution and performance analysis," Ultralytics Documentation, 2025. [Online]. Available: https://docs.ultralytics.com.

[51] Y. Zhao *et al.*, "DETRs beat YOLOs on real-time object detection," RT-DETR Project Page, 2025. [Online]. Available: https://zhao-yian.github.io/RTDETR/.

[52] Roboflow, "RF-DETR: Real-time object detection and segmentation model," GitHub Repository, 2025. [Online]. Available: https://github.com/roboflow/rf-detr.

[53] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015.

[54] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[55] M. Caccia, “Autonomous surface craft: Prototype and basic research issues,” *IFAC Proc. Volumes*, vol. 39, no. 2, pp. 91–96, 2006.

[56] A. Vasilijević, T. Petrović, and N. Mišković, “Multi-sensor fusion for autonomous underwater navigation and object detection,” *J. Field Robot.*, vol. 36, no. 5, pp. 798–814, 2019.

[57] NOAA Marine Debris Program, “Autonomous surface vehicle for marine debris detection and removal,” NOAA Technical Report, 2021.

[58] Y. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Comput. Electron. Agric.*, vol. 147, pp. 70–90, 2018.

[59] S. Norouzzadeh *et al.*, “Automatically identifying, counting, and describing wild animals in camera-trap images,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 115, no. 25, pp. E5716–E5725, 2018.

[60] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[61] G. Jocher, “Ultralytics YOLOv5,” 2020. [Online]. Available: https://github.com/ultralytics/yolov5.

[62] Y. Zhao *et al.*, “DETRs beat YOLOs on real-time object detection,” *arXiv preprint arXiv:2304.08069*, 2023.

[63] X. Yuan *et al.*, “A survey of target detection and recognition methods in underwater turbid areas,” *Appl. Sci.*, vol. 12, no. 10, Art. no. 4898, 2022.

[64] NOAA Marine Debris Program, “Autonomous surface vehicle for marine debris detection and removal,” NOAA Technical Report, 2021.

[65] The Ocean Cleanup, “The Ocean Cleanup project overview,” 2022. [Online]. Available: https://theoceancleanup.com/.

[66] A. Vasilijević, T. Petrović, and N. Mišković, “Multi-sensor fusion for autonomous underwater navigation and object detection,” *J. Field Robot.*, vol. 36, no. 5, pp. 798–814, 2019.

[67] Coral Triangle Initiative (CTI-CFF), “Coral reef monitoring using autonomous vehicles,” CTI-CFF Report, 2021.

[68] J. E. Manley, “Unmanned surface vehicles, 15 years of development,” in *Proc. IEEE OCEANS 2008*, Quebec City, QC, Canada, 2008.

[69] S. Wang, X. Zhao, and Y. Tang, “Deep learning algorithms for autonomous surface vehicles: Recent advances and applications,” *IEEE Access*, vol. 8, pp. 10492–10504, 2020.

[70] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, “Unmanned surface vehicles: An overview of developments and challenges,” Annual Reviews in Control, vol. 41, pp. 71–93, 2016.

[71] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[72] N. Carion *et al.*, “End-to-end object detection with transformers,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[73] Y. Zhao *et al.*, “RT-DETR: Real-time detection transformer,” Project Page, 2025. [Online].

[74] Roboflow, “RF-DETR: Real-time object detection and segmentation,” GitHub Repository, 2025. [Online].

[75] A. Vasilijević, T. Petrović, and N. Mišković, “Multi-sensor fusion for autonomous underwater navigation and object detection,” *J. Field Robot.*, vol. 36, no. 5, pp. 798–814, 2019.

[76] NOAA Marine Debris Program, “Autonomous surface vehicle for marine debris detection and removal,” NOAA Technical Report, 2021.

[77] A. Verma and B. Gupta, “Edge computing in autonomous surface vehicles: A real-time data processing approach,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5098–5108, 2020.

[78] M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning,” in *Proc. 12th USENIX Symp. Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

[79] F. Chollet *et al.*, “Keras,” GitHub repository, 2015.

[80] A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[81] G. Bradski and A. Kaehler, *Learning OpenCV*, O’Reilly Media, 2008.

[82] J. Bai *et al.*, “ONNX: Open neural network exchange,” GitHub repository, 2019.

[83] Tencent, “NCNN: High-performance neural network inference framework optimized for mobile platforms,” GitHub repository, 2025.

[84] Intel Corporation, “OpenVINO toolkit,” 2025.

[85] SeaClear Consortium, “SEarch, identification and Collection of marine Litter with Autonomous Robots (SeaClear),” Horizon 2020 Project No. 871295, CORDIS, European Commission. [Online]. Available: https://cordis.europa.eu/project/id/871295.

[86] SeaClear Consortium, “SeaClear Deliverable D2.3: Search, identification and collection of marine litter with autonomous robots,” Dec. 2021. [Online]. Available: https://seaclear-project.eu.

[87] J. Hong, M. Fulton, and J. Sattar, “TrashCan: A semantically-segmented dataset towards visual detection of marine debris,” *arXiv preprint arXiv:2007.08097*, 2020. [Online]. Available: https://arxiv.org/abs/2007.08097.

[88] M. S. Fulton, J. Hong, and J. Sattar, “Trash-ICRA19: A bounding box labeled dataset of underwater trash,” 2020.

[89] M. Pedersen, D. Lehotský, I. Nikolov, and T. B. Moeslund, “BrackishMOT: The brackish multi-object tracking dataset,” *arXiv preprint arXiv:2302.10645*, 2023. [Online]. Available: https://arxiv.org/abs/2302.10645.

[90] Aalborg University Visual Analysis and Perception (VAP), “The Brackish Dataset.” [Online]. Available: https://vap.aau.dk/the-brackish-dataset/.