IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 October 2023, accepted 6 November 2023, date of publication 9 November 2023,
date of current version 21 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332032

RESEARCH ARTICLE

Integration of Stereo Vision and MOOS-IvP for Enhanced Obstacle Detection and Navigation in Unmanned Surface Vehicles

YOUSEF ABD ALHATTAB, ZULKIFLI BIN ZAINAL ABIDIN, AHMED RIMAZ FAIZABADI, (Student Member, IEEE), HASAN F. M. ZAKI, AND AHMAD IMRAN IBRAHIM, (Member, IEEE)

Department of Mechatronics, Kulliyah of Enginnering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia
Centre for Unmanned Technologies (CUTe), International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia

Corresponding author: Zulkifli Bin Zainal Abidin (zzulkifli@iium.edu.my)

This work was supported in part by the Ministry of Science, Technology, and Innovation (MOSTI) through a Collaborative Effort Between Pesaka Mawar Sdn Bhd and International Islamic University Malaysia (IIUM); in part by the Research Management Centre; in part by the Project Autonomous Surface Vessel for Sedimentation Investigation In Port under Grant RD04201444 (eDana) and Grant SPP20-027-0027 (IIUM); and in part by the Obstacle Detection and Collision Avoidance for Unmanned Surface Vessel in Dynamic Environments'' Project led by the Centre for Unmanned Technologies (CUTe), IIUM.

ABSTRACT This paper addresses the development of a stereo vision-based obstacle avoidance system using MOOS-IvP for small and medium-sized Unmanned Surface Vehicles (USVs). Existing methods predominantly rely on optical sensors such as LiDAR and cameras to discern maritime obstacles within the short- to mid-range distances. Nonetheless, conventional cameras encounter challenges in water conditions that curtail their effectiveness in localizing obstacles and planning paths. Furthermore, LiDAR has limitations regarding angular resolution and identifying objectness due to data sparsity. To overcome these limitations, our proposed system leverages a stereo camera equipped with enhanced angular resolution to augment situational awareness. The system employs recursive estimation techniques to ascertain the position and dimensions of proximate obstacles, transmitting this information to the onboard control unit, where MOOS-IvP behaviour-based software produces navigation decisions. Through the real-time fusion of data obtained from the stereo vision system and navigational data, the system is able to achieve Enhance Situational Awareness (ESA) and facilitate well-informed navigation decisions. Developing a state-of-the-art maritime object detection technique, the system adeptly identifies obstacles and swiftly responds via a vision integration protocol. During field tests, our system proves the efficacy of the proposed ESA approach. This paper also presents a comprehensive analysis and discussion of the results derived from deploying the proposed system on the Suraya Surveyor USV platform across numerous scenarios featuring diverse obstacles. The results from these various scenarios demonstrate the system's accurate obstacle detection capabilities under challenging conditions and highlight its significant potential for safe USV operations.

INDEX TERMS Stereo vision, object detection, situational awareness, obstacle avoidance, MOOS-IvP, unmanned surface vehicles.

I. INTRODUCTION

Unmanned Surface Vehicles (USVs) have gained a large amount of attention in recent years due to the advancement of computing and sensor technologies. This traction is because USVs can undertake tasks that are safety-critical missions such as surveillance and reconnaissance [1], environmental monitoring [2], and hydrographic surveys [3]. Collision avoidance is a vital capability, particularly for USVs, which cannot benefit from the real-time guidance of a human operator. Safe maritime navigation and collision avoidance remain the main two challenges since they require the

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo.

seamless coordination of multiple complex subsystems [4]. Safe operations of USVs encompass various aspects such as obstacle detection, path planning, guidance, and control. In order for autonomous navigation systems to operate effectively, it is crucial for vessels to possess the capability to perceive and comprehend their surroundings across various environments.

The automatic detection of surrounding objects and accurate motion estimation are crucial elements that require reliable and robust performance across diverse environmental conditions [5]. Therefore, an additional sensing approach is necessitated for Enhance Situational Awareness (ESA) in no Global Navigation Satellite System (GNSS) zone without Automatic Identification System (AIS) or reliable Global Positioning System (GPS) communication. In addition, marine radars are low resolution and slow in sampling, affecting high-speed or small object detection. This situation can be perceived in USVs by using one or more additional sensors like cameras or infrared systems. The vessel must also react quickly and intelligently to avoid this set of known targets, following the required needs of path planning, guidance, and control [6]. Unfortunately, many USVs remain unable to perform one or more of these crucial tasks, limiting their adoption beyond the oceanographic research community. In maritime traffic of uncooperative or conducive environments, information on surrounding vessels can be obtained from optical sensors such as cameras that have been successfully used for many robotics applications. They can detect close-range obstacles at a relatively high sampling rate. However, conventional cameras are marred by limited visibility in water conditions, lighting variations, and limited depth perception. Furthermore, localizing objects concerning SA and path planning is non-trivial.

In this paper, we propose a robust object detection model for ESA and safe USV navigation using stereo cameras. Stereo cameras have a relatively high angular resolution. Thus, they can enhance the detectability of short- or mid-range targets [7]. The obtained three-dimensional (3D) data from the stereo vision consist of arranged corresponding points, which helps to collect surrounding obstacle information within a wide environmental view. Integrating such sensors into USV navigation and control can lead to an ideal solution to improve target detection performance compared to the conventional radar-only-based approach that needs to be investigated [8]. Furthermore, this paper also discusses the field test results achieved using a collision avoidance method based on a stereo vision system with the Mission Oriented Operating Suite-Interval Programming (MOOS-IvP) [9]. The field test approach used during the verification exercise deviates from the typical approach used in research experiments, where test scenarios are usually limited to fixed pre-planned behaviours of obstacles with specific path planning. This work is implemented on the newly developed USV platform ''Suraya-Surveyor'' [10]. The Suraya-Surveyor USV system incorporates a suite of detection sensors consisting of LiDAR, an electro-optical (EO) camera, and a stereo camera system which is the scope of this study. The results show that our system can detect obstacles accurately, even in challenging conditions such as low light and turbidity at slow variable speeds. In summary, there are four significant contributions of this paper in advancing the frontier of USV research. These include:

1) Arobust object detection model for safe USV navigation, using a federated dataset that enhances the detectability of short- to mid-range obstacles.
2) A software module employing stereo cameras to extract 3D data and information on nearby obstacles enhances situational awareness.
3) A complete protocol for integrating the software module to utilize object detection and collision avoidance for USV behaviours using MOOS-IvP.
4) A field test evaluation to benchmark the system performance for MOOS application in challenging scenarios.

The remainder of the paper is organized as follows. Section II provides an overview of related works. Section III outlines the methods for developing the USV platform, an object detection module, software for SA, and the Vision Integration Protocol (VIP) is discussed in Section IV. The MOOS-IvP implementation is shown in section V, followed by the presentation of results and field experiments in Sections VI. The discussion in section VII highlights the impacts of the proposed approach, shortcomings and suggests areas for future work. Finally, the paper concludes with a conclusion.

II. RELATED WORK

Extensive studies have advanced SA strategies in recent years. This section explores integrating stereo vision systems into USV's autonomous navigation and collision avoidance systems. Optical sensors, like cameras and LiDAR, detect and track obstacles at close range, while stereo vision systems provide precise target state estimates. The applications, advantages, and challenges of these sensing technologies in USVs are discussed, highlighting advancements in autonomous navigation and collision avoidance

A. OPTICAL SENSORS FOR SITUATIONAL AWARENESS

USVs require autonomous navigation and collision avoidance in close-range marine conditions for safe and efficient operations. A significant amount of research has been undertaken on autonomous navigation and collision avoidance techniques for USV platforms. The popular approach to deal with SA in the absence of GPS-based systems and to overcome the limitations of marine radar systems is to use optical sensors like LiDAR and cameras [7]. In marine environments, cameras are commonly deployed as sensors for detecting and tracking close-range obstacles. In the context of bearing-only tracking, a monocular camera offers bearing information, while the estimation of relative range information can be indirectly inferred from the changes in

bearing. Feature-based image processing techniques were applied to detect targets on camera images, and their motions were estimated by tracking filters [11]. However, relying on a single camera to obtain the precise three-dimensional location of a target is non-trivial.

3D LiDARs utilize vertically arranged laser beams to gather information on surrounding obstacles by rotating the lasers for a complete 360° view. While LiDAR provides precise relative bearing and range data for close-range obstacles, it has limitations in angular resolution and object classification. To overcome these limitations, combining cameras and LiDAR sensors has been explored for detecting close-range obstacles in marine environments [4], [12].

Stereo vision systems are crucial for precise target state estimation of close-range objects in marine environments [13]. They are used in detecting dynamic and static obstacles in close range for marine vehicles [14]. However, their performance relies on textured surfaces for robust stereo matches. Muhovic et al. [7] used an onboard stereo camera and IMU to approximate the water surface accurately in calm seas. Nevertheless, their stereo reconstruction falls short in detecting partially submerged obstacles, and increasing the stereo baseline for distant obstacles negatively impacts USV stability. Bovcon et al. [15] extended a semantic segmentation method based on fitting structured models to RGB images [16]. Their approach utilized a stereo camera system and an IMU sensor to enhance water-edge estimation and reduce false-positive detections caused by environmental factors. However, this method struggles with the diversity of marine scenes, leading to poor segmentation in the presence of visual objects like wakes, sea foam, glitter, and reflections.

One of the most recently developed stereo vision systems is the ZED camera which has two high-resolution cameras. According to the manufacturer's information, it is designed and built to perceive the depth of objects in indoor and outdoor environments [17]. Recent studies have suggested using ZED cameras as it returns reliable depth values [18]. ZED was employed in many studies for a sensing subsystem of short-term obstacle detection and avoidance functionality of the USV, which is, in turn, part of the more complex automatic onboard navigation system [12], [19]. For the perception of the vehicle, ZED and 3D LiDAR were used [12], and the data from these two sensors were processed, analyzed, and used for collision decision-making. However, these strategies and methods are computationally expensive, and the methodologies have not been tested in experimental non-controlled environments.

Therefore, a better use case for ZED to utilize its full potential for improvised SA is needed. Table 1 presents a comparison of studies utilizing optical sensors to provide valuable insights. The sensor capabilities must be augmented with a robust AI model for object detection in a maritime environment. This enhancement will significantly improve the detectability of short- or mid-range targets for small and medium-sized USVs. Additionally, a well-curated dataset for marine object detection is imperative for the development of a robust model for short-term collision avoidance. The effectiveness of the collision avoidance system is heavily reliant on a robust object detection model, which is paramount for safe USV navigation. While a classical feature-based detection algorithm was initially employed for vision-based obstacle detection using camera images [11], [20], it demonstrated reasonable performance in well-configured and steady illumination conditions. However, the parameter tuning of the feature-based detection algorithm proves to be rather challenging and sensitive to real-world varying environmental conditions.

B. DEEP LEARNING-BASED MARITIME OBJECT IDENTIFICATION

A deep-learning-based strategy has been used to increase the robustness and reliability of marine obstacle identification, utilizing successful applications such as the Single Shot MultiBox Detector (SSD) [21]. SSD is a convolutional network that detects objects by producing bounding boxes and class scores. It consists of a base network and extra feature layers. The base network, a 16-layer model, provides classification results. The extra feature layers at the back help detect object locations of different sizes. Finally, a loss value is computed using classification and detection scores from each feature layer. Besides its computational efficiency and accuracy in detecting large objects, SSD has limitations in detecting and localizing small objects, which can be attributed to the default anchor box sizes and the limited receptive field in early network layers [22].

One of the best-known single-stage object detectors is YOLO (You Only Look Once) [23]. This architecture addresses object detection as a regression problem to obtain spatially separate bounding boxes and associated class probabilities. In one evaluation, a single neural network directly predicts bounding boxes and class probabilities from full images. YOLO has demonstrated superior performance to previous state-of-the-art methods, such as SSD [24], attributed to its unified network architecture that directly predicts bounding boxes and class probabilities, effectively capturing contextual information. YOLO's grid-based approach and architectural improvements like feature pyramid networks and attention mechanisms contribute to its exceptional capability in detecting small objects. These factors make YOLO a preferred choice for applications requiring accurate detection of small objects, including maritime objects. In addition, these algorithms have enhanced the performance of vision-based detection in USV platforms [25], [26].

The advantages of using a deep learning algorithm lie in the quality and quantity of the data fed into the model during the training process. Many datasets, such as SeaShip [27] and Singapore Marine datasets (SMD) [28], have been created for maritime object detection. SeaShip is a large-scale ship dataset featuring high-resolution images of six ship types, capturing diverse scenarios with variations in ship types, scales, viewpoints, illumination, and occlusion

TABLE 1. Comparison of camera-based approaches for collision avoidance in USVs.

Study	Camera-based distance	Small Object Detection	Collision Avoidance	ESA	Remarks
Monocular Camera [11]	No	No	No	Poor	Provides basic SA information and precise 3D target location is not available
Camera & LiDAR Sensors [12]	No	Yes	Yes	Good	Computationally expensive and tested in controlled environments
Stereo Camera & IMU [7, 15, 16]	Yes	Yes	No	Average	Poor segmentation as struggling with diverse marine scenes, reducing SA effectiveness
ZED2i [19]	Yes	Yes	No	Not Applied	Limitations in testing and data availability
This work	Yes	Yes	Yes	Very Good	3D target estimation, power efficient and handles diverse scenes with exceptional SA performance

degrees [27]. However, SeaShip exclusive use of side views from offshore cameras makes it suboptimal for collision avoidance. SMD encompasses various classes related to the maritime environment, including vessels, maritime infrastructure, navigation aids, marine equipment, and coastal landscapes [28]. However, SMD may not fully capture the variability and complexity present in an actual maritime environment. Therefore, a critical need for a large-scale dataset encompassing diverse data poses (front, isometric views) and scales (i.e., very near objects) of vessels and boats to develop an effective vision system for collision avoidance, enhancing SA for USVs.

C. MARITIME COLLISION AVOIDANCE AND PATH PLANNING

Field experiments validate the performance of maritime situational awareness and obstacle avoidance systems. Studies addressing obstacle avoidance have been conducted and verified with field experiments for the autonomous operation of marine vehicles in maritime traffic situations [4], [29]. With USV systems developed on commercial kayak platforms, a study of automatic collision avoidance using a monocular camera was conducted [20]. Moreover, a collision-avoidance experiment was performed using a model predictive control [30], and a velocity obstacle-based collision avoidance algorithm was tested on a maritime combat vessel [31]. Nevertheless, with an increase in the number of target objects, the algorithm's computational complexity increases, which affects the decision-making time and leads to losing the tracked objects. Hepworth et al. [32] have utilized a stereovision-based navigation system for collision avoidance to provide an appropriate solution for autonomous inland vessels only. However, these are single-layer approaches, where one algorithm solves the complete collision avoidance problem, which might cause the re-planned path information to be lost.

One approach to address path planning challenges is to use action selection and multi-objective optimization [33]. This architecture enables layered behaviours, activating complex ones when specific conditions are met. It allows for emergent behaviours and incorporates a multi-objective solver with reduced computational overhead. MOOS-IvP, released in 2006, brokers a compromise between competing objectives, resembling a ship captain's decision-making process to reach waypoints while avoiding collisions and hazards [34]. Table 2 presents the comparison of these studies. It is evident from the literature that efficient SA can improvise the safe operations of USVs. We aim to build a robust maritime object detection model and a powerful stereo-vision sensor for improved situational awareness. The proposed situational awareness algorithm is integrated with a robust control framework for multi-object optimizations for collision avoidance, and the complete framework is tested thoroughly through field experiments.

TABLE 2. Collision avoidance approaches in USVs.

Study	Approach	Limitations
[29]	A local reactive obstacle avoidance approach	Lacks global map data; Limiting path optimization
[20]	Autonomous collision avoidance using a monocular camera	Challenging 3D target localization
[30]	A model predictive control-based collision avoidance	Used for collision avoidance with COLREGs behaviours
[31]	Velocity obstacle-based collision avoidance	Increased complexity with more targets can lead to losing tracked objects
[32]	Stereovision-based navigation system	Specifically designed for autonomous inland vessels
[34]	Action selection and multi-objective optimization	Initially designed to work in conjunction with LiDAR
This work	Integration of Stereo Vision and MOOS-IvP	focus on the detectability of short- to mid-range obstacles; Uses stereo camera-based approache for collision avoidance

III. PROPOSED METHODOLOGY

In this section, we present the methodology employed to achieve the four major contributions of this work. Firstly, we describe the system overview and development of the USV platform, including the hardware and software. Then, we discuss the method of developing a robust vision-based object detection model for safe USV navigation. Particularly,

this model uses a federated dataset to improve the detectability of obstacles at close and intermediate ranges. In addition, the model is built to efficiently recognize and keep track of items of interest, which ensures reliable capabilities to avoid collisions. Next, using the proposed federated data model, we construct a stereo based module to extract 3D data and acquire precise information about adjacent obstructions. This module improves SA by giving a complete picture of the environment

Furthermore, we propose a complete protocol for integrating the software module into the USV system. This protocol ensures MOOS-IvP uses, object detection for USV collision avoidance systems. Likewise, integrating the module into the USV control system enables real-time decision-making based on accurate obstacle detection and avoidance behaviours. Finally, to evaluate our system's performance, we conducted field testing of the complete MOOS application. These tests evaluate the system's capabilities in various operating circumstances, revealing its real-world applicability and performance. Through the methodology described in this section, we intend to develop a robust and reliable framework for object detection and collision avoidance to improve the safety and effectiveness of USVs navigation.

A. SYSTEM OVERVIEW

Figure 1 illustrates the architecture of the USV system, highlighting the essential components of the Onboard System (OBS) and the remotely linked Ground Control Station (GCS). The Enhance Situational Awareness (ESA), which comprises the perception module, the Vision Integration Protocol (VIP), and the computing model, provides the seamlessly integrated controller with enhanced object details and pertinent information. Obstacles are effectively identified by leveraging stereo camera ZED2i [35], a notable member of the ZED cameras family. Subsequently, ZED2i captures frames, enabling a deep-learning model to detect potential objects. Each detected object is assigned a unique identification, and VIP is updated with new measurements for each tracked object for data transmission.

The computing module receives vital information from navigational sensors, including the compass, Inertial Navigation System (INS) and Real-Time Kinematic Global Positioning System (RTK-GPS) data. It is also used as a host for serial device server, enabling data transmission to multiple onboard processing units. This data is accessible at GCS to facilitate remote control decisions for manual intervention for failed safe operations.

The autonomous decision-making module processes decisions based on filtered and estimated positions of tracked objects, which are further assessed for subsequent processing. Navigation commands are then generated through various MOOS applications, providing crucial information about the current position and trajectory of the USV platform. This approach facilitates behavior-based autonomy, enabling the hierarchical development of complex missions. Also, within MOOS-IvP, a mathematical programming technique is employed to seek globally optimal solutions within each domain, resolving conflicts between different behaviors and ensuring the continuous update of desired navigation data.

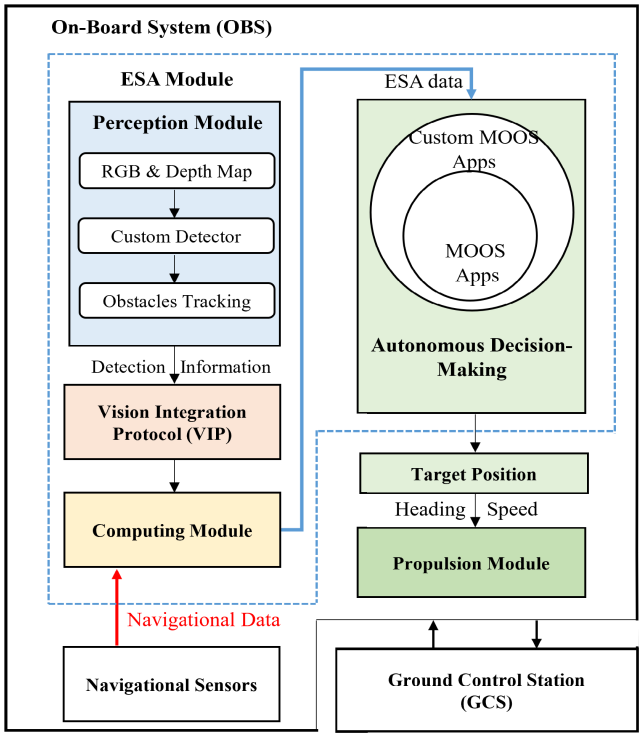


FIGURE 1. Block diagram of System Architecture and Components.

1) USV PLATFORM

To ensure the development of a stable platform capable of accommodating future research applications, we built Suraya Surveyor USV shown in Figure 2, which is a monohull boat. Suraya Surveyor, measuring 4.7 meters in length and 1.4 meters in width, has been specifically designed for coastal bathymetry and hydrology purposes. Equipped with a waterjet propulsion system, diverse navigation and perception sensors, communication systems, and processing units, the integrated USV system offers comprehensive functionality. Detailed specifications of Suraya Surveyor can be found in Table 3. Notably, this platform has an enhanced payload capacity compared to its predecessor. Additionally, it utilizes multi-beam technology in its survey system, allowing for more precise mapping capabilities while operating at speeds ranging from 2 to 8 knots, with endurance of 8 and 2 hours respectively.

Table 3 provides the specifications of Suraya Surveyor system. The system has multiple payload capabilities, allowing it to perform various roles. The transducer used in the system has nine degrees of beamwidth and enables effective sensing and data acquisition. For positioning, the Suraya Surveyor utilizes the Atlaslink H-10 positioning system, which offers a high level of accuracy with an RMS

FIGURE 2. Suraya- Surveyor Unmanned Surface Vehicle.

TABLE 3. Suraya Surveyor specifications.

Specifications	Description
Payload	Multirole system with multiple payload capabilities
Transducer	Airmar 9° beamwidth
Positioning	Atlaslink H-10 positioning (<8cm RMS Accuracy) as standard
Telemetry	Fail-safe telemetry via Long Range WiFi, Kongsberg MBR, Sailor Ku Band Satellite Broadband
Tracking	Iridium Short Burst Data (SBD) worldwide asset tracking
Acquisition Software	Beamworx, NAVAQ
Propulsion	Dual thruster and rudderless propulsion
Remote Control	Remote with autonomous mission waypoint function
Payload Options	Scalable payload options

accuracy of less than 8cm. Telemetry is ensured through fail-safe connectivity options, including Long Range WiFi, Kongsberg MBR, and Sailor Ku Band Satellite Broadband. The system utilizes Iridium Short Burst Data (SBD) for worldwide asset tracking). The acquisition software used is Beamworx. The propulsion system of the Suraya Surveyor consists of dual thrusters and is designed to operate without a rudder. The system can be remotely controlled and includes an autonomous mission waypoint function. Additionally, Suraya Surveyor offers scalable payload options, allowing for customization and flexibility in payload selection.

2) HARDWARE SYSTEM

Figure 3 depicts the hardware architecture of the obstacle avoidance system on the USV platform, which comprises navigation and perception sensors, a propulsion system, and a computing module.

The Navigation sensors are equipped with an RTK-GPS, a compass (KVH C100) [36], and an INS (SBG Systems Ellipse-E) [37] to provide accurate and reliable vehicle motion information. The SBG Systems Ellipse-E is a small device with a high-performance inertial navigation system. It consists of a MEMS-based inertial measurement unit (IMU). Moreover, it utilizes an enhanced extended Kalman filter (EKF) to combine inertial and aiding information to produce precise real-time orientation and navigation data. This device can provide data at a maximum rate of 1000 Hz. In addition to the Eclipse-E, the system incorporates a microprocessor-controlled fluxgate compass, KVH C100, which consists of a detachable toroidal fluxgate sensor element and a compact electronics board. C100's innovative automatic compensation algorithm allows it to handle tilt angles up to 450, with a resolution of 0.1 and an advertised accuracy of 0.5. This cost-effective

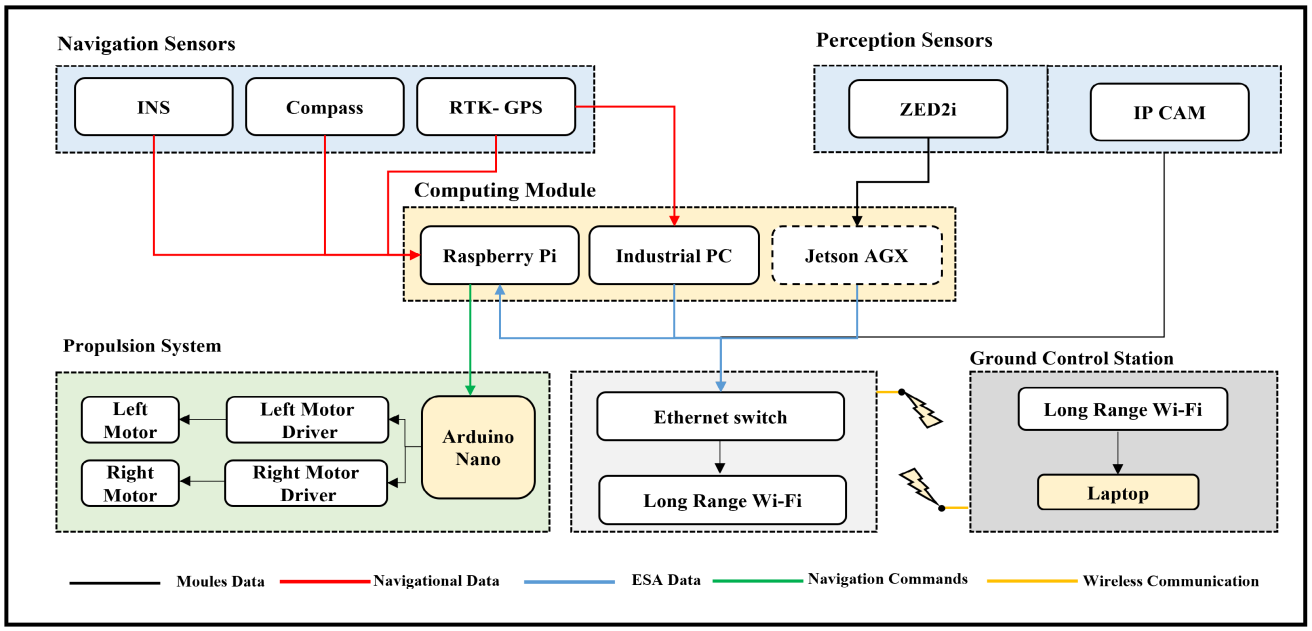


FIGURE 3. Hardware Architecture of Suraya Surveyor Obstacle Avoidance System.

system contributes significantly to achieving high accuracy levels.

Furthermore, the system utilizes the onboard ZED2i to perceive and recognize external obstacles and conditions. The camera data is accessed through Ethernet communication, and its data is transmitted directly to the computing module for automatic obstacle detection. The IP camera stream is transmitted to the ground station, where a detection and tracking method is implemented on the operator's laptop for detection data collection. The Arduino Nano is interfaced with the Raspberry Pi to precisely control the two motors with the desired heading and speed.

The computing module, housing a Raspberry Pi and an industrial PC, collects and processes navigation data from these sensors. Simultaneously, GPS data is also directed to the industrial PC for bathymetry and hydrology data collection. The computing module is interconnected via an Ethernet hub, enabling it to exchange data through User Datagram Protocol (UDP) ports. NVIDIA Jetson Xavier AGX handles the processing of detection and tracking data, leveraging its exceptional performance in demanding graphical and arithmetic functions, including the deep learning models employed in this work. The AGX is connected to the ZED2i via USB 3.0 and to other navigation sensors through Ethernet. It executes all algorithms developed in this work using JetPack 4.4, which features L4T 32.4.3.

Maritime broadband radio system (MBR) was implemented to establish communication between the ground station and Suraya-Surveyor systems. Specifically, we utilize MBR 179 (as shown in Figure 4) [38], which is a Maritime Broadband Radio developed for use in maritime applications where reliable communication and data transfer are critical

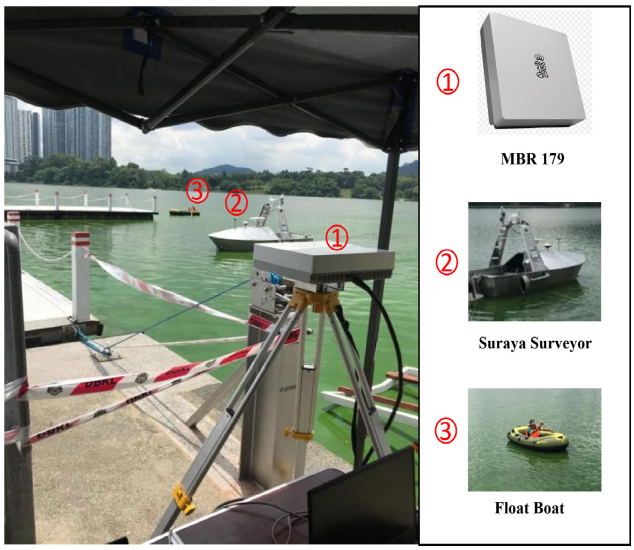


FIGURE 4. On-site MBR 179 Broadband Radio System.

for efficient and safe operations. This system allows for digital high-speed data transfer, ensuring that data can be transmitted and received quickly and reliably.

3) SOFTWARE SYSTEM

The software system architecture for the autonomous obstacle avoidance system is shown in Figure 5. The onboard sensors provide their measurements in the sensor referenced frame. Therefore, navigation data is utilized to transform the sensor measurements to MOOS-IvP and determine the obstacle's location and USV state.

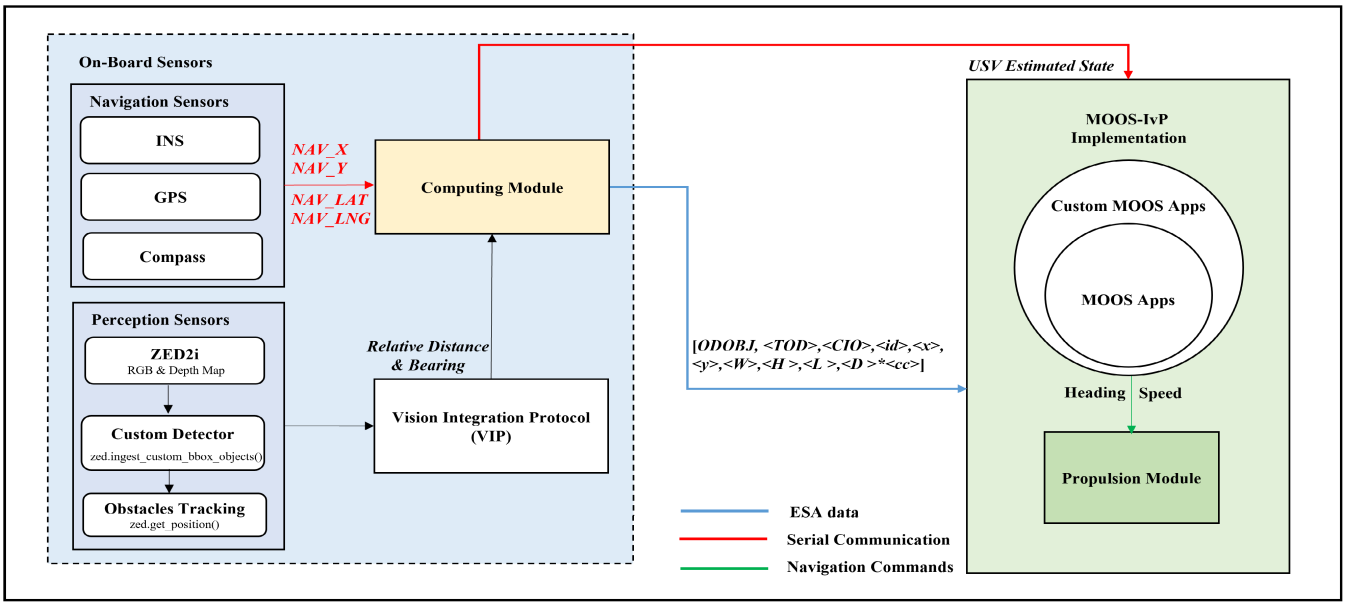


FIGURE 5. Software architecture of Suraya Surveyor obstacles avoidance system.

The detection and tracking module incorporates a custom detector model utilizing PyTorch [39] and YOLOv5 [40] for processing RGB images. In addition, the detector runs within an edge device, enabling real-time results to broadcast via UDP within a local network. This simplifies device communication and allows for easy function and data sharing. Once the custom detector has detected the objects, the detected objects of interest are identified, and detection information is generated using ZED APIs for distance and 3D dimension of an object. Then, the relative distance and bearing are being transmitted to the computing module to generate the ESA data while keeping updating the USV state.

The ESA data is fed into the controller, which then only requires the application of MOOS behavior on the ESA data to achieve the desired heading and speed. The MOOS behavior relies on the MOOS application set, which can execute commands quickly through a communication channel that forms the edge of the MOOS-IvP subsystem. Typically, a MOOS process manages this channel. To provide feedback, the interface process periodically requests the current position, speed, and heading from the communication channel. At the conclusion of each iteration, the chosen MOOS behaviour orders the desired heading and speed, resulting in the present state of the vehicle (position, speed, and heading).

B. FEDERATED MARITIME DATASET

While existing open-access datasets are valuable for identifying multiple static targets, there are notably lack of specificity required for precise maritime detection. Therefore, we have created a new Maritime Federated Large Dataset (MFLD) by combining COCO [41], SeaShip, and SMD to address the limitation mentioned for available datasets in Section II. In addition, several processing steps were implemented to ensure a cohesive dataset.

To ensure cohesiveness and usefulness, the methodology for creating MFLD involved several crucial steps. Firstly, the consolidation of vessel and ship classes into a single class using the in-house developed Class Mapper. This step streamlined the dataset and facilitated more manageable object detection tasks. Additionally, the data labels were converted to the annotation schema format using the labels converter, further enhancing compatibility and ease of use. When dealing with dataset that require video processing such as SMD, a systematic approach was implemented. Concretely, every fifth frame was extracted from each video in the SMD, ensuring a representative sample while minimizing redundancy. Four selected video frames were included in the test set, while the remaining videos contributed only their fifth frame. This consistent approach was applied across the three categories in the dataset: infrared, onshore, and onboard footage.

The converter component was responsible for converting the data labels to the annotation schema format suitable for object detection tasks. It processed the file format containing the annotations and extracted relevant information such as class names, bounding box coordinates, and image dimensions. This information was then used to generate a unified framework format in the desired annotation format, which could be efficiently utilized for training object detection models.

Extracting the boat class from the COCO enhanced the dataset's diversity and content. This inclusion introduced a more comprehensive range of boat samples, contributing to a more comprehensive representation of maritime objects. By incorporating this additional data, the dataset became

TABLE 4. Proposed federated large datasets comparative.

Dataset	SMD	Seaship	MFLD
Purpose	Maritime environment	Ship object detection	Maritime environment
Modality	NIR, RGB	RGB	NIR, RGB
Published Format	Videos	Images	Videos, images
Annotation Format	XML	Darknet	unified
Number of Imagest	6350	7000	16375
Object Categories	Various maritime classes	Six ship types	Various maritime classes
Image Resolution	High	High	High
Data Variability	Diverse maritime scenarios	Diverse ship scenarios	Diverse maritime scenarios
Percentage of Ship/Boat Images	53%	100%	100%

more versatile and capable of addressing various real-world scenarios. In parallel, the mapper component was crucial in consolidating the various vessel and ship classes into a single class. It iterated through the extracted class names and grouped similar classes together under the unified label. This mapping process ensured consistency and coherence within the dataset, simplifying subsequent analysis and model training. Combining the converter and mapper components transformed the initial dataset into a cohesive and standardized format. The dataset became more streamlined and consistent by converting the labels to the appropriate annotation schema and consolidating the vessel and ship classes. These processing steps were fundamental in creating a unified and well-structured dataset that could effectively support object detection tasks in the maritime domain.

Figure 6 shows that proposed MFLD is curated by collecting 3025 images from COCO for boat class. A collection of 7,000 images from SeaShip and 6350 images from SMD total of 16375 boat images. The training images are selected to include various environmental and weather conditions, which enables robust obstacle detection of the network.

The process of creating MFLD is illustrated in Figure 6, providing a visual representation of the dataset creation. Also, in Table 4, we provide a summary of the proposed MFLD datasets, including their purpose, data variability, number of images, and labels. This comparative analysis highlights the complementary aspects of the combined federated dataset, showcasing its ability to provide broader coverage of maritime object detection challenges.

C. VISION-BASED OBSTACLE DETECTION MODEL

This work employs a fast, reliable, deep-learning-based detection algorithm for robust and reliable target obstacle detection. YOLOv5 is used considering both its computational efficiency and detection accuracy after some preliminary performance comparison of the SSD object detection method in our applications. YOLOv5 is a real-time object detector that builds on the work done on the YOLO family of object detectors. It consists of a base network and an extra feature layer. The backbone module extracts feature from the input image based on Focus, Bottleneck CSP (Cross Stage Partial Networks), and SPP (Spatial Pyramid Pooling) and transmits them to the neck module. The neck module generates a feature pyramid based on the PANet (Path Aggregation Network). It enhances the ability to detect objects with different scales by fusing low-level spatial features and high-level semantic features bidirectionally. The head module generates detection boxes, indicating the category, coordinates, and confidence by applying anchor boxes to multiscale feature maps from the neck module. Through multiple studies, YOLOv5 has demonstrated its effectiveness in maritime object detection, consistently achieving high accuracy [26], [42]. An appropriate dataset and input image size are essential for training a neural network. Therefore, we employ a set of different YOLOV5 model training. First, the network is trained with 640 $\times$ 640 input images and uses MFLD and active learning. For our application, YOLOv5 models that we explored in the work were: YOLOv5n, YOLOv5s, YOLOv5m, and YOLOv5l. The network initialization parameters are shown in Table 5.

TABLE 5. Initialization parameters of YOLOv5 model.

Image size	Batch	Momentum	Learning rate	Decay
640 x 640	16	0.9	0.001	0.0005

YOLOv5n represents the most minor variant of YOLOv5. It prioritizes inference speed and is suitable for real-time or near-real-time detection tasks where speed is crucial. YOLOv5s is slightly larger than YOLOv5n and strikes a balance between model complexity and inference speed. It offers improved detection accuracy while maintaining reasonably fast performance. YOLOv5m is a medium-sized model that further enhances detection accuracy compared to YOLOv5s. It is suitable for scenarios where higher precision is desired, even if it comes at the cost of a slightly slower inference speed. YOLOv5l is the largest variant among the YOLOv5 models. It offers the highest level of detection accuracy but requires more computational resources for

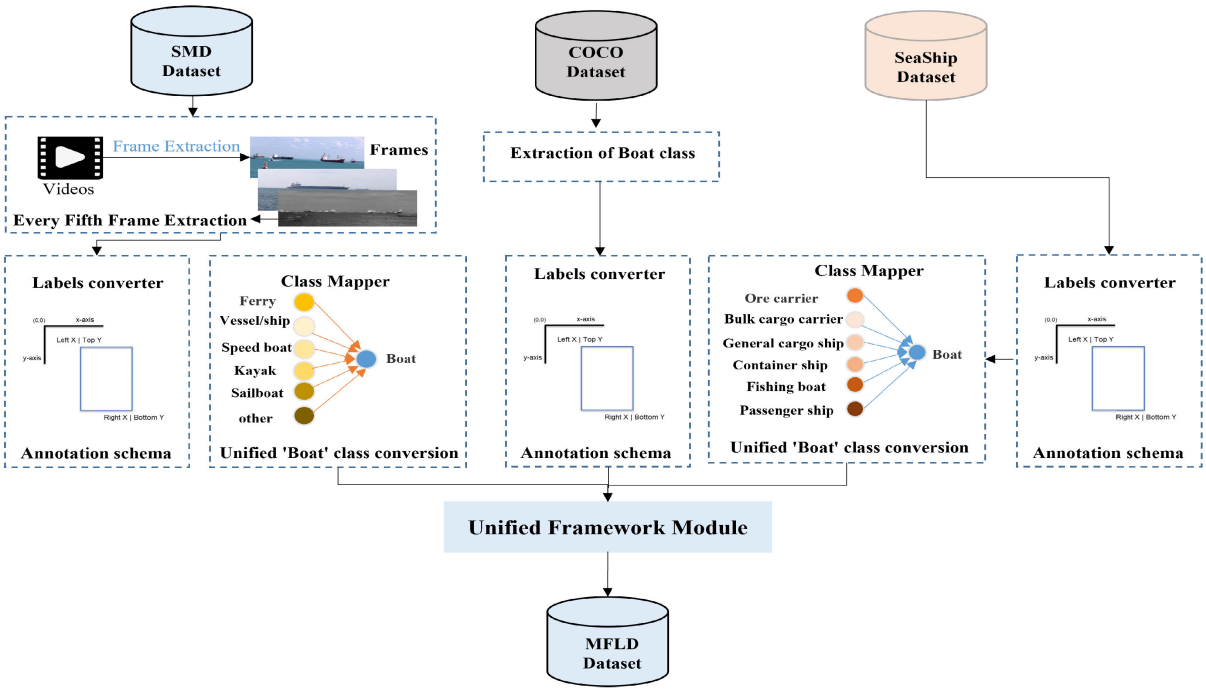


FIGURE 6. Proposed Methodology for Creating Maritime Federated Large Dataset.

TABLE 6. YOLOv5 models average Frames-per-Second (FPS).

Model	YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l
FPS	60*	54	37	19

inference. It is recommended for tasks that prioritize accuracy over speed.

It is essential to consider the frame rate cap when evaluating the performance of YOLOv5 networks with ZED2i. The frame rate cap represents the maximum number of frames processed per second by the model. In our experiments, we applied a frame rate cap during the testing phase to assess the real-time inference capabilities of the YOLOv5 networks.

The system operates at 30 frames per second with a resolution of 1080HD, setting the standard for object detection and tracking parameters. Therefore, it is imperative that the YOLOv5 model can handle processing within the range of this FPS value of FPS. Table 6 provides an overview of the average frames-per-second (FPS) achieved by different YOLOv5 models, highlighting the impact of model size on the system's frame rate capabilities, which is reported by [40].

The algorithmic procedure of the system is extensively presented in ESA Algorithm (Algorithm 1), as following:

As depicted in Algorithm 1, the ESA algorithm initializes the ZED2i camera object, sets configuration parameters, and initializes tracking parameters. It captures the point cloud data and RGB image from the left camera. The algorithm then enters a loop, creating a detection thread and ingesting custom bounding box objects using the ZED2i camera

Algorithm 1: Enhance Situational Awareness Algorithm

Input: 3D and RGB data
Output: object dimensions and id
initialize the zed camera object
set configuration parameters
initialize tracking parameters
capture point cloud
capture RGB (left)
while img_available and Not exit_signal **do**
    create detection_thread()
        //custom bbox format converter
    objects ← *zed.ingest_custom_bbox_objects(detections)*
    m ← *zed.retrieve_measure()*
    D ← *Estimate_depth()*
    i ← *zed.retrieve_image()*
    p ← *zed.get_position()*
    for all objects **do**
        ESA_msg ← *prepare_ESA_string(m, i, p)*
    end for
    return ESA_msg to parent
end while

object. It retrieves measurements, estimates depth, retrieves images, and obtains positions. For each object, it prepares an ESA message by combining the measurements, image, and position. Finally, it returns the prepared ESA message to the parent or calling function.

The overall methodology is depicted in Figure 7 which depicts the image capture module simultaneously capturing RGB and depth map images. First, the depth map is obtained

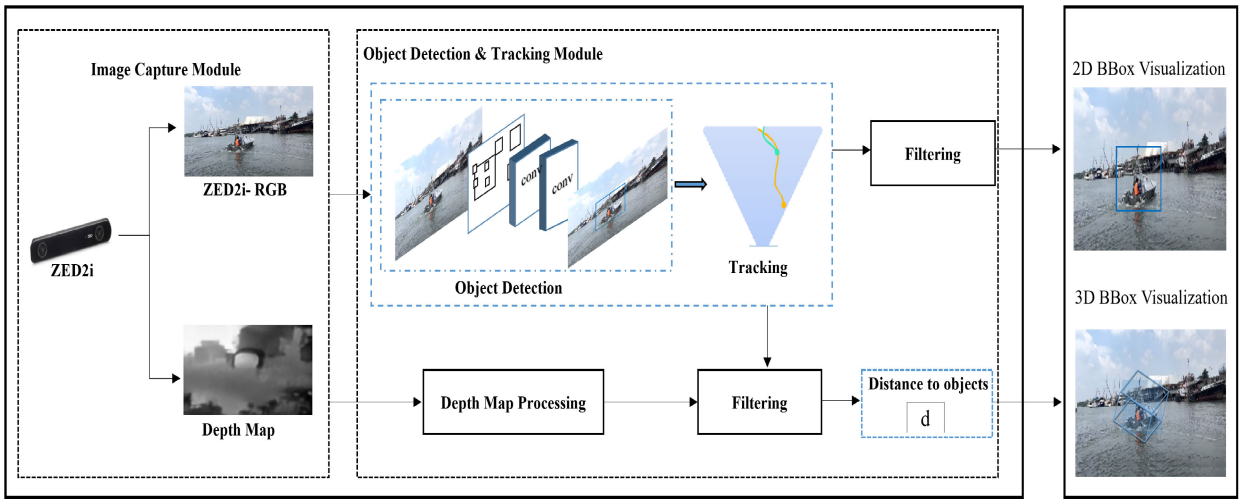


FIGURE 7. Methodology for ESA using Object Detection and Tracking using ZED2i.

using stereo vision principles, where the rectified calibration matrix ensures alignment between the left and right images. This alignment enables the calculation of depth information based on the disparity between corresponding pixels in the stereo image pair.

Next, the object detection and tracking module, implemented using the ZED API, identifies potential objects in the captured frame. A custom maritime model detector developed using MFLD is also used for improving maritime object detection. After processing the depth map, a filter is updated for each tracked object using the new depth measurements, which estimate the object's state based on the observed depth information. Subsequently, based on the filtered position of the tracked objects, the distance is obtained. Finally, bounding boxes are commonly used to denote their location in 2D and 3D space to visualize objects in the frame.

D. ZED DEPTH ESTIMATION

ZED2i captures RGB images and depth data, enabling object detection and distance calculation. RGB images are analysed using YOLOv5 custom model for maritime objects and ZED API with a pretrained model for other objects identification, while depth data determine object distance based on averaged depth values. Furthermore, object localization is enhanced with 3D estimate of object. ZED2i has a max range of 20 meters [35]. ZED2i computes depth information using triangulation (re-projection) from the geometric model of non-distorted rectified cameras. Assuming that the two cameras are co-planar with parallel optical axes and the same focal length $f_l = f_r$, the depth Z of each point L is calculated by equation 1, here B is the baseline distance and $\mathrm{xi}^l - \mathrm{xi}^r$ is the disparity value [43]. Notice that depth varies inversely proportional to the disparity.

$$Z = \frac{fB}{\mathrm{xi}^l - \mathrm{xi}^r} \tag{1}$$

ZED2i cameras acquire 3D values considering the left camera as the origin frame. Given the coordinates of a pixel (u', v') in this image coordinate frame, these exact coordinates are used to search their corresponding depth value Z in the depth map. Furthermore, depth maps store the distance value Z for each pixel (x, y) present in the image. The distance is expressed in meters and is calculated using the equation (2) that measures the distance between the back of the left eye of the camera and the object in the scene.

$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{2}$$

where P_1 is the distance in pixels between the centroid of the object and the end of the overlap area for the camera on the left. P_2 is the distance in pixels between the centroid of the object and the beginning of the overlap area for the right camera.

IV. VISION INTEGRATION PROTOCOL

To ensure seamless integration of the ESA and designed hardware with a controller, a module for communication and surveillance over the network has been developed, utilizing the client-server architecture. As illustrated in Figure 8, the Vision Integration Protocol (VIP) module forms a critical component of the approach described in this section using client, server architecture for better integration. The perception sensor feeds the data to Inference Engine for object detection and tracking running on Jetson Xavier AGX and interacts with GPS receivers, INS, and digital compass, while simultaneously sending and receiving data in the computing module. Additionally, the detection module provides the pixel position of the centre of the detected objects' bounding boxes, enabling the calculation of the positions, and determining locations of objects.

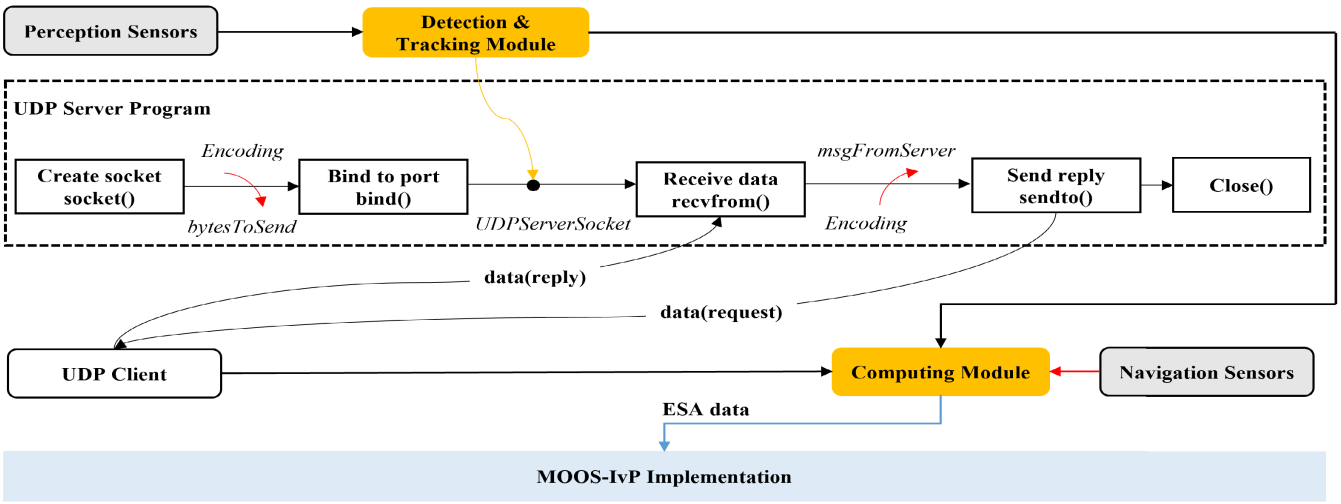


FIGURE 8. VIP for Collision Avoidance Approach.

The code converts bounding box coordinates from the format (x_1, y_1, x_2, y_2) to the format (x, y, w, h), as the following function:

$$x, y, w, h = xyxy2xywh(x_1, y_1, x_2, y_2) \tag{3}$$

where x represents the x-coordinate of the top-left corner, y represents the y-coordinate, w represents the width, and h represents the height of the bounding box.

Then, to convert normalized xywh coordinates to bounding box corners (A, B, C, D), we use the function xywh2abcd(x, y, w, h, im[0].shape). This function takes x, y, w, h values of the bounding box along with the shape of the image (im[0].shape) and returns the coordinates A, B, C, D representing the four corners of the bounding box, as shwon in (4):

$$A, B, C, D = xywh2abcd(x, y, w, h, im[0].shape) \tag{4}$$

The analysis module processes the detected object data from ZED2i. The analysis module also incorporates navigation data from the navigation sensors, allowing for the combination and refinement of object and navigational data. The extracted 3D information and tracking details for each object is called ESA message. The resulting data is then formatted into navigational data for the controller.

VIP comprises of server-client module responsible for transmitting data over the network for better integration with any controller in use. The algorithm for Vision integration Protocol (VIP) system is illustrated in Algorithm 2 as follows:

The system defines object and tracking parameters at 30 frames per second and 1080HD resolution, captures images, and detects potential objects in the scene. It then identifies the location of these objects using a set of parameters for controller processing, which includes colour images, and depth maps. The UDP server-client module

Algorithm 2 : Algorithm VIP Server

Input: ESA_msg <detections, objects>
Output: Encoded ESA_msg (EM) //NMEA sentence
socket(type)
bind(ip,port) //Bind to address & IP
for object in objects **do**
  for detection in detections **do**
    //Get object dimensions & object position
    if distance is not NaN and not Infinite **then**
      *print_log(**ESA**_message)*
      MS ← *prepare_client_msg*(**ESA**_*message*)
      EM ← *Encode*(**MS**)
      *Sendto(**EM**, address)*
    end if
  end for
end for

uses the following National Marine Electronics Association (NMEA) format for each object detected in the scene.

$ODOBJ, <TOD>,<CIO>,<id>,<x>,<y>,<W>,<H>, <L>,<D><cc>CRLF*

Table 7 provides a detailed overview of the NMEA format for object data. This format condenses object information into a single line of ASCII text, employing commas as separators. Each line is constrained to 80 characters. The initial field is denoted by the code *$ODOBJ*, with$x$ and y indicating the object's top-left origin in meters

The format entails a crucial parameter in the form of object count detected in the scene, which serves as vital input for the data collection system of the base station. This information aids in making operational decisions and providing feedback to the controller. The checksum (cc) is calculated with an XOR operation of all characters between '$' and '*'. VIP passes an NMEA sentence, returns the NMEA data, and

TABLE 7. NMEA format description.

Model	Description
$	Start delimiter
ODOBJ	Message ID
TOD	Number of objects detected in the frame
CIO	Current object index, starts with 1
id	Object ID
x	X coordinate of the origin in meters
y	Y coordinate of the origin in meters
W	Object width in meters
H	Object height in meters
L	Object length in meters
D	Distance to object
*	Checksum delimiter
cc	Checksum value in hexadecimal (0x00 to 0xFF)
CRLF	Newline character

is appended to the sentence when generated and locally calculated. Then, NMEA sentences are read and processed to be used by a controller. Algorithm 3 shows the NMEA sentence checksum calculation, this entire encoding forms a Encoded ESA message generated by VIP server.

Algorithm 3 :NMEA Sentence Checksums Calculation

Input: NMEA sentence as a string
Output: checksum: Calculated checksum
def nmea_checksum(st):
    $i \leftarrow 0$
    checksum $\leftarrow 0$
while $i <$ length of *st* **do**
        checksum $\leftarrow$ *checksum* $\oplus$ ord($st[i]$)
        $i \leftarrow i + 1$
end while
return *checksum*

Algorithm 3 calculates the checksum for an NMEA sentence by defining a function called *nmea_checksum* that takes the detection data as the input string. This function initializes two variables, *i* and *checksum*, to zero and uses a while loop to iterate over each character in the string. Within the loop, the ordinal value function is used to get the ASCII code of the current character, and then the *XOR* operator is used to update the checksum variable. After the loop finishes, the final value of the checksum is returned. This calculated checksum can then be appended to the NMEA sentence and transmitted between devices to ensure the data's integrity.

The Encoded Message (EM) is prepared using the Object Tracking module's ESA message, as shown in Algorithm VIP Client (Algorithm 4). It then creates a socket of a specified type. The algorithm binds the socket to a given IP address and port. In an infinite loop, the algorithm proceeds to send the byte stream of the EM to the server's IP address using the specified port. It then waits to receive a message (msg) from the server. Upon receiving the message, the algorithm decodes it, resulting in a Decoded Message (DM). The DM is then sent to the controller or another designated recipient using the Send_controller function. This loop continues indefinitely, allowing for continuous communication between the VIP Client and the server

Algorithm 4 Algorithm VIP Client

Input: EM (Encoded ESA Message), Server Port Address
Output: Navigation commands
Initialize buffer
socket(type)
bind(IP, port)
while True **do**
    Sendto(bytetosent, Server_IP)
    Msg $\leftarrow$ *Recvfrom()*
    DM $\leftarrow$ *Decode(**Msg**)*
    *Send_controller(**DM**)*
end while

V. ACTION-SELECTION AND MULTI-OBJECTIVE OPTIMIZATION

Due to its compliance with the autonomy requirements outlined in Section II, this work utilizes MOOS-IvP, as the chosen autonomy framework for inter-process communications. MOOS-IvP works in a star topology without peer-to-peer communication and is governed by the MOOSDB using a publish-subscribe architecture, as shown in Figure 9. This architecture enables independent, easily replaceable MOOS apps.

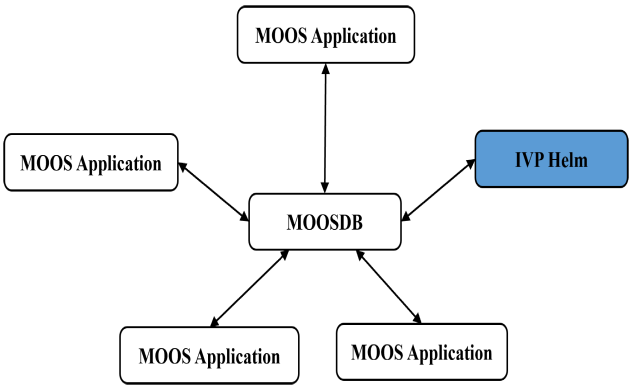


FIGURE 9. A MOOS community.

MOOS applications typically run on a single machine with a separate process ID. Each process communicates through a single MOOS database (MOOSDB) in a publish-subscribe manner. In addition, each process may execute its inner loop at a frequency independent from one another and set by the user. For example, MOOS-IvP is a MOOS application designed to provide autonomy on robotic platforms and is

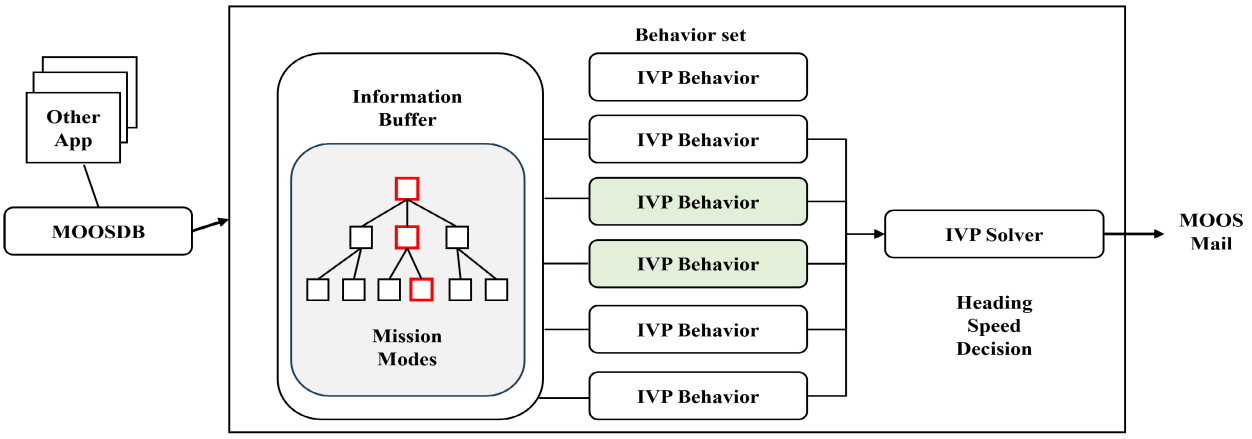


FIGURE 10. IvP Helm Autonomy Loop Architecture [44].

particularly well-suited to marine vehicles. In Figure 10, IvP behaviours are depicted as they determine how the vehicle responds to its environment in pursuit of a defined goal [44].

IvP functions are a mathematical programming model comprising the piecewise linear representation structure and the solution algorithm that capitalizes on this structure. For a problem defined over a decision space with n decision variables $(x_1, \ldots, x_n)$, and having K objective functions $f_1(x_1, \ldots, x_n), \ldots, f_k(x_1, \ldots, x_n)$, with K priority weights $(w_1, \ldots, w_k)$, the general solution form is given by [44]:

$$x^* = \arg\max_x \sum_{i=1}^{K} (w_i \cdot f_i(x)) \tag{5}$$

Before vehicle deployment, a mission structure will be written in the form of helm behaviours and their configuration. The general mission structure is comprised of a set of mission modes D and a set of behaviours B:

$$M = \{D, B\} \tag{6}$$

The mode structure is comprised of M unique mission modes and the set of behaviors with B unique behaviors. Each mode is defined by a unique name and set of logic conditions, whereas each behavior is defined by a unique name and set of configuration parameters for obstacle avoidance [44].

A. OBSTACLE AVOIDANCE MANAGER

The basic layer of obstacle avoidance in MOOS-IvP involves *pObstacleMgr* [45] application and the *AvoidObstacleV21*. As shown in Figure 11, detected objects are represented as convex polygons with *IDs* and managed by *pObstacleMgr*. This behaviour creates a buffer zone around each obstacle, assigning a negative cost to headings that intersect with it, guiding the vehicle towards a safer path. The helm uses heading and speed information to navigate safely around obstacles [44].

The buffer obstacle may dynamically and temporarily shrink if the USV position intrudes, shrinking until the USV is no longer inside. Then, as the USV opens range, the buffer

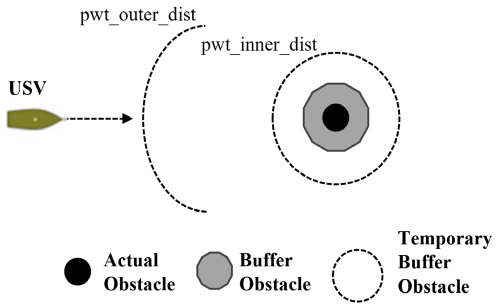


FIGURE 11. Defined Convex Polygon Around the Actual Polygon Obstacle [44].

TABLE 8. Behaviour parameters configuration.

Parameters	Configuration	Parameter Labels
buffer_dist	3	Buffer Distance
pwt_inner_dist	10	Priority Weight Inner Distance
pwt_outer_dist	20	Priority Weight Outer Distance
allowable_ttc	5	Time-to-Collision

obstacle will be re-grown back to its original size, producing a heading-speed decision. The behaviour parameters are being configured with the following sets showed in Table 8:

Where `buffer_dist` is a nonnegative distance, in meters, from which to make a buffer polygon around the actual obstacle. `pwt_inner_dist` is a range to the obstacle, in meters, within the behaviour will have maximum priority weight, and `pwt_outer_dist` is the range to the obstacle, in meters, beyond which the behavior will have a zero-priority weight. Finally, `allowable_ttc` is the number of seconds beyond which a manoeuvre that steers the robot on a collision course to an obstacle is acceptable [44].

B. MOOS APPLICATIONS DEVELOPMENT

In our collaborative MOOS application development, we've integrated new applications to support a newly

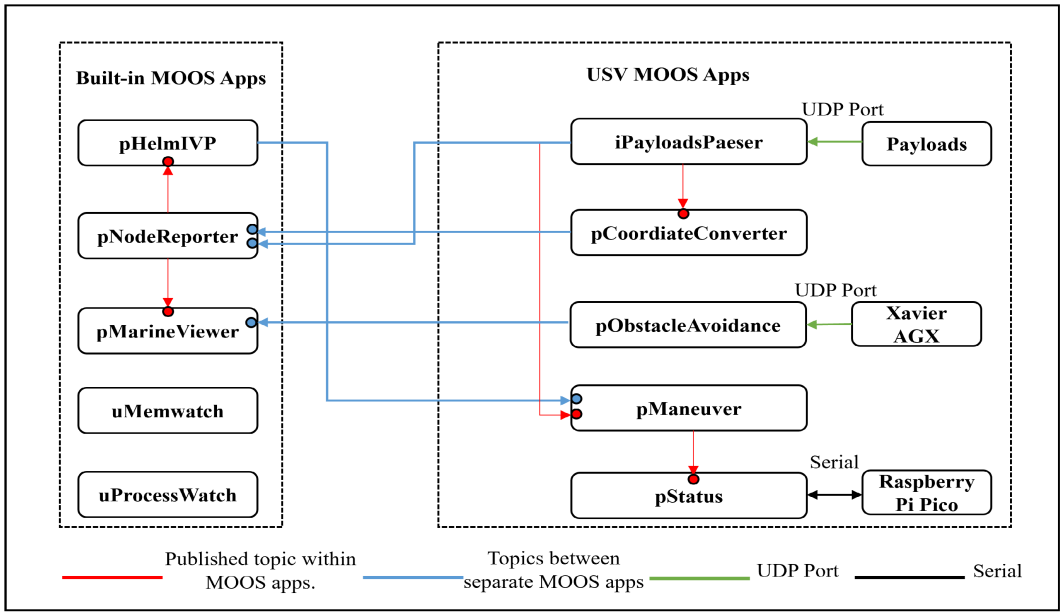


FIGURE 12. MOOS Publish-Subscribe Communication Module Architecture Detailing Applications and Connectivity Mechanisms.

TABLE 9. Published and subscribed topics of MOOS apps.

MOOS app	pHelmIvP	pNodeReporter	pMarineViewer	uMemWatch	uProcessWatch
Subscribe topics	MOOS_MANUAL _OVERRIDE	NAV_X NAV_Y NAV_LAT NAV_LNG NAV_HEADING NAV_SPEED	NOOE_REPORT_LOCAL VIEW_POINT	DB_CLIENTS	DB_CLIENTS
Publish topics	DESIRED_HEADING DESIRED_SPEED	NOOE_REPORT_LOCAL VIEW_POINT	-	PHELMIVP_MEM PNODEREPORTER_MEM	-

developed USV. These applications enable autonomous operation in dynamic environments. We began by configuring existing MOOS apps and then developed new ones.

1) BUILT-IN MOOS APPLICATIONS

Table 9 shows the published and subscribed topics for each MOOS application.

1) **pHelmIvP** [44]: is a behaviour-based autonomous decision-making MOOS application. It consists of a set of behaviours reasoning over a common decision space, such as the vehicle heading and speed. Behaviours are reconciled using multi-objective optimization with the IvP model. pHelmIvP receives the Note_Report_Local, which comes from pNodeReporter, and calculates the desired destination for the USV (Unmanned Surface Vehicle). Subsequently, it publishes the desired heading and speed primarily.
2) **pNodeReporter** [46]: Creates a custom message to be consumed by the pMarineViewer app and displays the vehicle. It contains the latitude and longitude information (LAT/LNG), heading, and speed with the XY of the vehicle.
3) **pMarineViewer** [47]: This provides a quick and easy-to-custom interface to visualize vehicles and other variables
4) **uMemWatch** [48]: application used for measuring the current memory used by a set of MOOS apps.
5) **uProcessWatch** [48]: It monitors the presence of a set of MOOS applications and Central Processing Unit (CPU) load of a set of MOOS applications. Table 9 shows the published and subscribed topics for each MOOS application.

2) USV PLATFORM MOOS APPLICATIONS:

Table 10 presents the published and subscribed topics for each MOOS Application of the USV platforms.

1) **iPayloadsParser**: it handles the raw payload data sent by the devices and sensors in order to set initial values and extract the measured variables. It is mainly responsible for publishing the navigation data such as LAT/LNG.

TABLE 10. Custom MOOS apps topics.

MOOS app	iPayloadsParser	pCoordianteConverter	pObstacleAvoidance	pManeuver	pStatus
Subscribe topics	NAV_LAT NAV_LNG	NAV_X NAV_Y NAV_HEADING	NAV_X NAV_Y NAV_HEADING	DEPLOY RETURN LOITER DISABLE_THRUSTERS NAV_HEADING NAV_SPEED DESIRED_HEADING DESIRED_SPEED CONTROLLER_GAIN THROTTLE_VALUE	DRIVER_RESPONSE NAV_LIGHTS
Publish topics	NAV_HEADING NAV_SPEED	NAV_X NAV_Y	VIEW_POINT NAV_LIGHTS	VESSEL_TYPE SERVO_PULSE_VALUE DRIVER_RESPONSE	CPU_TEMPERATURE BATTERY_VOLTAGE

2) **pCoordianteConverter**: it uses for the most part of converting from LAT/LNG to XY, and handles missing data from pNodeReporter; also, it loops the subsequent iterations for filling in missing LAT/LNG information.
3) **pObstacleAvoidance**: this app implements obstacle detection and tracking method. It will receive a specific command from the onboard AI edge device via User Datagram Protocol (UDP) ports, which allows the system to differentiate the received sensing data of different sensors. It also published the tracked features, which contain the information on detected and tracked obstacles to XY of the vehicle. This updates the data for manoeuvring the vehicle to avoid obstacles as known locations, expressed as one or more convex polygons. The safe-distance tolerance and policy for priority based on the range is provided in the mission configuration.
4) **pStatus**: includes app-specific status message configuration, runtime warnings, and notable events. It is connected to the microcontroller Raspberry Pi Pico which monitors and alerts on battery voltages and CPU temperature.
5) **pManeuver**: it publishes navigation information to MOOSDB. It deploys Loiter behaviour with the current heading and speed. It merely expresses a preference for a particular thruster's speed and heading, and the vessel type and values of servo pulse are published in this app as well.

The MOOS applications integrated into the USV platform facilitate communication and exchange of information. These built-in applications interact with their counterpart MOOS applications, ensuring real-time data sharing, as depicted in Figure 12.

VI. RESULTS AND FIELD EXPERIMENT

This section presents the findings obtained from the testing and validating different steps of the proposed methodology. First, the section will discuss the outcomes of training different YOLOv5 models using diverse datasets. Subsequently, the results of field experiments conducted under multiple scenarios will be presented.

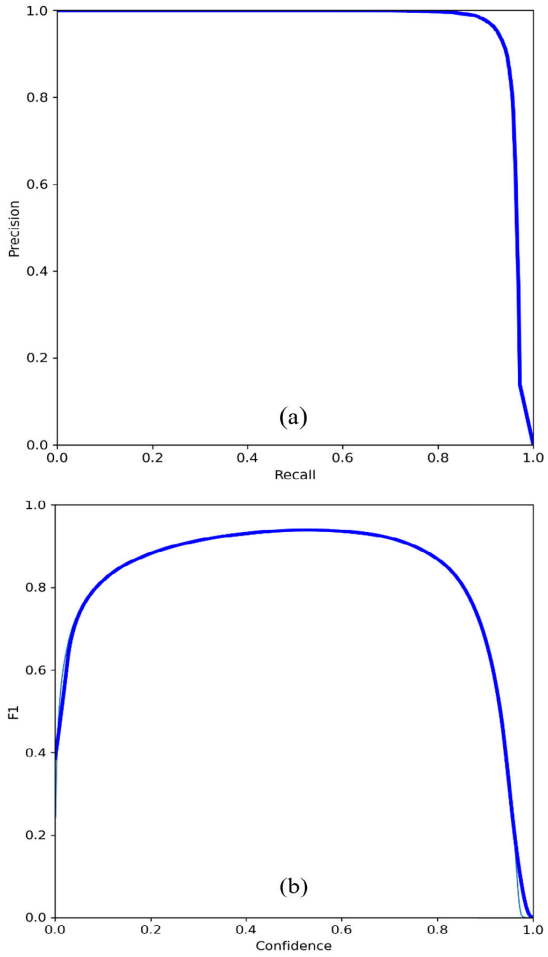


FIGURE 13. (a) YOLOv5s F1 score curve, (b) YOLOv5s PR curve.

A. PRELIMINARY FINDINGS

Developing a robust vision-based object detection model for USV navigation is vital for ESA. Utilizing a federated dataset (MFLD) improves obstacle detectability, ensuring reliable collision avoidance. This comprehensive dataset overcomes limitations of individual datasets, providing diverse data for an effective vision system. The metric used for model

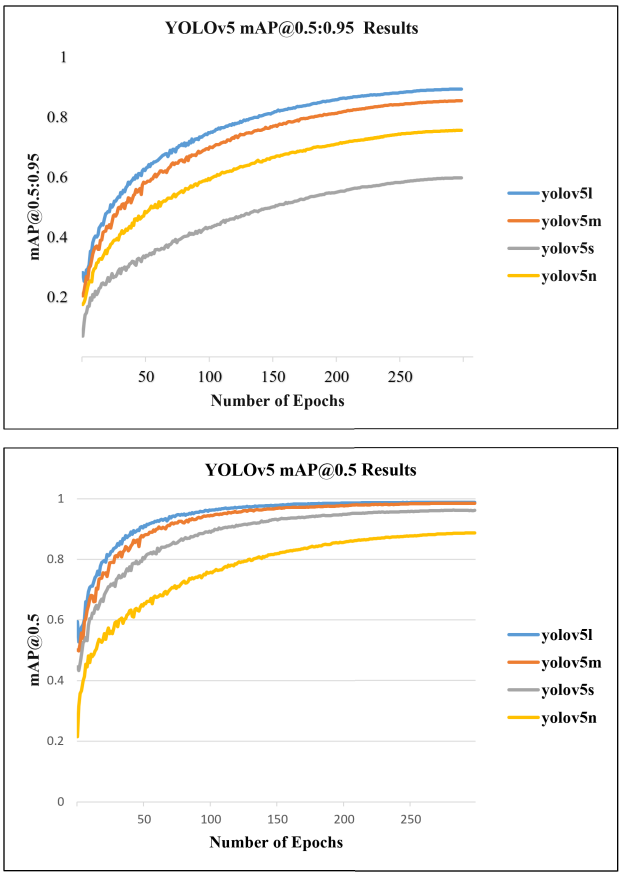


FIGURE 14. MLFD trainined Performance on Combined Evaluation Dataset.

evaluation will be *PR* curve and confidence score using *F1* as shown in Figure 13 for the YOLOv5s model. Model achieved an accuracy score above 96% with 300 epochs of training while maintaining real-time performance, effectively minimizing power and efficiently met the required number of FPS without compromising system responsiveness, ensuring smooth operation without any lag or delays.

The custom detector capabilities of the CNN variants, namely YOLOv5n, YOLOv5s, YOLOv5m, and YOLOv5l networks, were evaluated by training and testing them on MFLD. mAP results obtained from the four models' analysis for maritime objects within the ESA vision system for Suraya-Surveyor are presented in Figure 14. The evaluation results showed that YOLOv5l achieved highest precision (99.12%) and recall (97.55%) scores among the tested models, resulting in an overall F1 score of 98.32%. The model exhibited excellent performance in maritime object detection, as indicated by the mean Average Precision (mAP) scores at different IoU thresholds. However, FPS upper cap performance is also crucial to real-time navigation. Thus, trade-off between FPS cap and accuracy will determine the best model for use. mAP@0.5 of YOLOv5m and YOLOv5l are very close with 98.48% and 98.86% respectively.

Table 11 shows the average precision for each trained model. The metrics include precision, recall, *F1* score, and mean average precision (mAP) at different thresholds. The models show increasing levels of accuracy, with YOLOv5l achieving the highest precision, recall, and overall accuracy, followed by YOLOv5m and YOLOv5s. YOLOv5n performs slightly lower but still demonstrates respectable results.

TABLE 11. Average Precision, Recall, Precision, and F1 for trained models.

Model	Precision	Recall	F1 Score	mAP	
				@0.5	@0.5:0.95
YOLOv5n	90.02%	80.89%	85.27%	88.86%	59.80%
YOLOv5s	96.71%	91.62%	94.11%	96.23%	75.56%
YOLOv5m	98.73%	96.27%	97.48%	98.479%	85.45%
YOLOv5l	**99.12%**	**97.55%**	**98.32%**	**98.86%**	**89.36%**

The obtained results from the other single datasets for object detection using the YOLOv5s model at 30 FPS are summarized in Table 12. The table provides insights into the performance of the model when trained and tested on different datasets, with evaluations based on Precision (*P*), Recall (*R*), mean Average Precision at IoU 0.5 (mAP@0.5), and mean Average Precision at IoU 0.5 to 0.95 (mAP@0.5:0.95).

TABLE 12. Comparative analysis of YOLOv5s results utilizing model.

Training	Testing	P	R	mAP@0.5	mAP@0.5:0.95
SeaShip	SMD	0.19	0.2	0.1	0.01
SMD	SeaShip	0.205	0.281	0.113	0.0354
MFLD	SeaShip	0.667	0.47	0.493	0.236
MFLD	SMD	0.735	0.502	0.609	0.343

SeaShip tested in SMD, the precision was 0.19, indicating a relatively high rate of false positives. The Recall value of 0.2 suggests that the model missed many actual objects. The mAP@0.5 score of 0.1 implies that the model struggled to detect and localize objects accurately at an Intersection over Union (IoU) threshold of 0.5. Moreover, the mAP@0.5:0.95 value of 0.01 highlights the model's challenges in accurately capturing object boundaries across a wider range of IoU thresholds. Next, When the model was trained on SMD and tested on SeaShip, there were noticeable performance improvements. The precision increased to 0.205, suggesting a higher accuracy in object identification. The Recall value improved to 0.281, indicating a more remarkable ability to capture objects. The mAP@0.5 also increased to 0.113, signifying enhanced object detection and localization performance. Similarly, the mAP@0.5:0.95 rose to 0.0354, indicating improved precision across a wider range of IoU thresholds.

Training the model on MFLD with both SeaShip and SMD led to further improvements in performance. The precision increased to 0.667 and 0.735 for the SeaShip and SMD testing sets, respectively, indicating more accurate object identification. The Recall values improved to 0.47 and 0.502, respectively, suggesting a higher ability to capture objects.

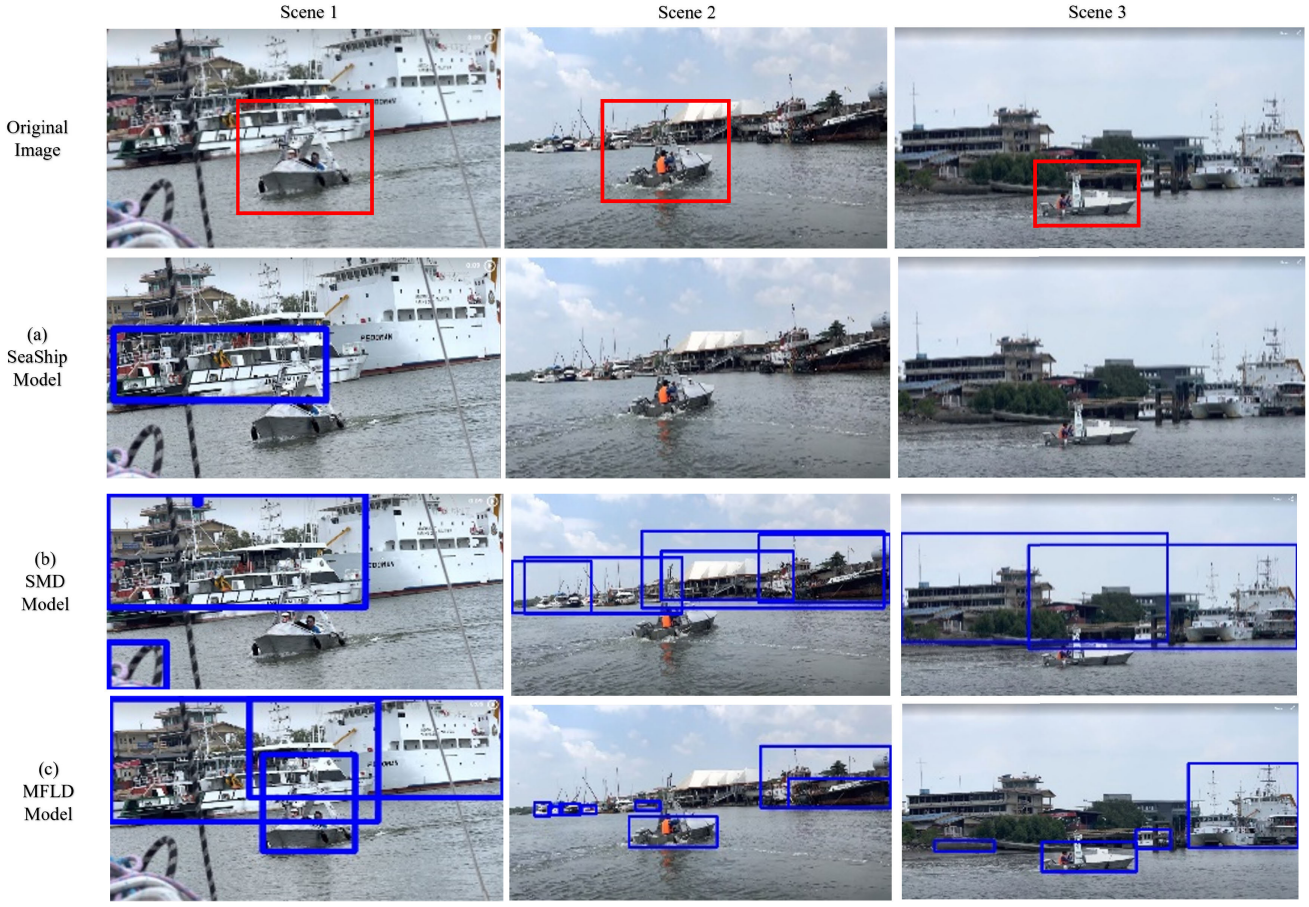


FIGURE 15. Presents detection on three distinct groups of images featuring the object of interest: (a) on SeaShip, (b) on SMD, and (c) on MFLD.

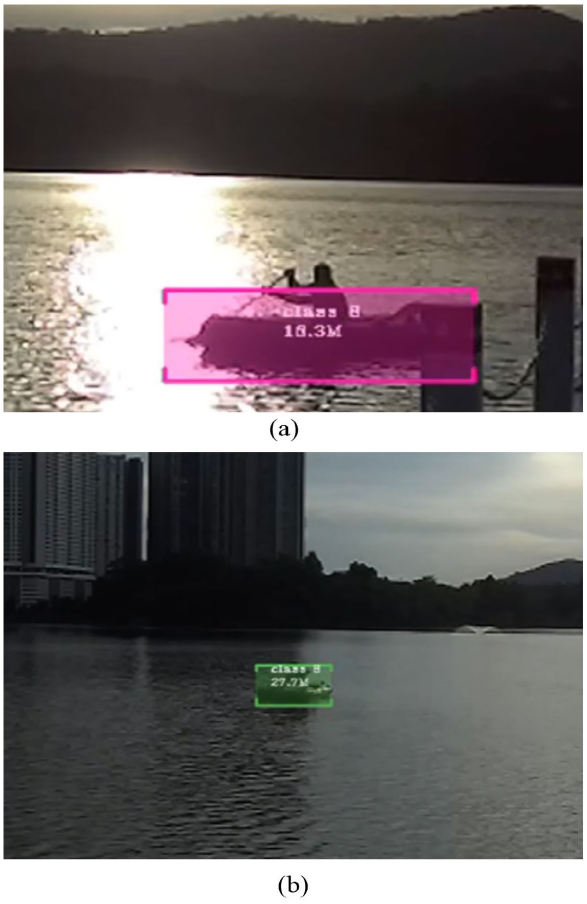
The mAP@0.5 values significantly increased to 0.493 and 0.609 for the SeaShip and SMD testing sets, respectively, indicating improved object detection and localization performance. Moreover, the mAP@0.5:0.95 values increased to 0.236 and 0.343, respectively, highlighting higher precision across a broader range of IoU thresholds. Figure 15 illustrates the detection results on different scenes from on-site.

The performance evaluation of the different models is also qualitatively compared. As seen in Figure 15, in the first scene, three objects are present. The MFLD model successfully detected all of them, while both SMD and SeaShip models missed two additional objects. However, SMD and Seaship models could not detect the object of interest, the closest vessel. Moving on to the second scene, a dozen objects are visible. The SeaShip model failed to detect any of them, while the SMD model displayed overlay bounding boxes and failed to identify the main object of interest accurately. Lastly, in the third scene, the MFLD model successfully detected all objects once again. However, both SeaShip and SMD models encountered issues, with SeaShip missing all objects and SMD overlaying bounding boxes without detecting the object of interest. In other words, with a clear sky view and proper illumination visible object of interest are not detected by SeaShip and SMD models in all three scenes. Whereas MFLD produces accurate detection of object of interest in all the scene proving its superiority over its predecessors.

TABLE 13. Omission rates in object detection.

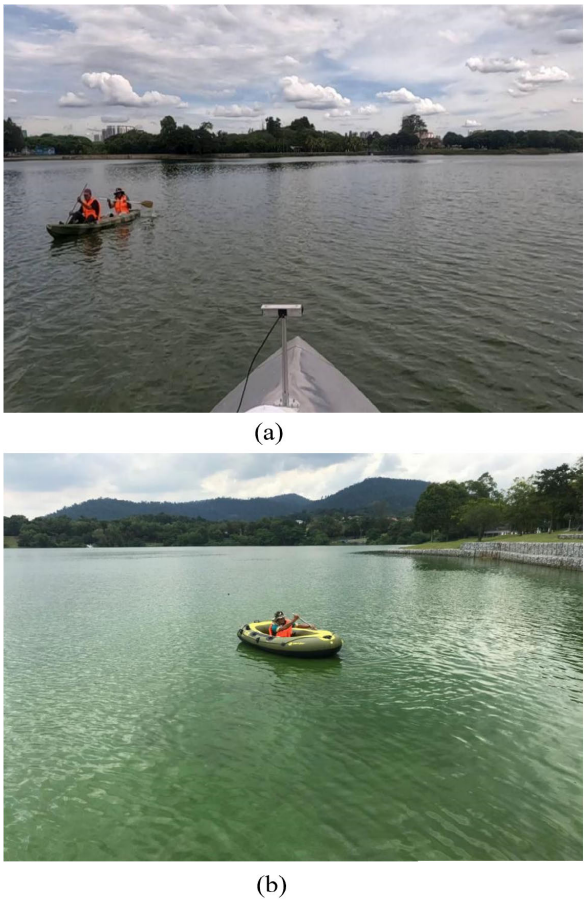
Training	Validation	Omission
SeaShip	SMD	0.8
SMD	SeaShip	0.72
MFLD	SeaShip	0.53
MFLD	SMD	0.5

The omission analysis was conducted to assess the performance of the YOLOv5s model in detecting objects across different training and validation dataset combinations. The omission rate represents the proportion of objects that the model did not detect. In the SeaShip-SMD scenario, the model was trained on SeaShip and validated on SMD, resulting in a high omission rate of 0.8. Training the model on SMD and validating it on SeaShip yielded a relatively lower omission rate of 0.72. This results showed in Table 13,



(a) (b)

FIGURE 16. YOLOv5s model detections with ZED2i.



(a) (b)

FIGURE 17. Varieties of Obstacles Used in the Experiments. (a) Kayak. (b) Floatable boat.

indicate a reduced proportion of missed objects compared to the SeaShip-Singapore scenario.

Moreover, additional performance enhancements were observed when the model was trained on MFLD with Seaship as the training set and validated on both SeaShip and SMD. The omission rate decreased to 0.53 in the SeaShip validation set, indicating a lower proportion of missed objects. Similarly, in the SMD validation set, the omission rate was 0.5, signifying a higher proportion of objects successfully detected by the model. The onsite, real-time low light challenging scenario of selected model on MFLD output is as shown in the Figure 16. Figure 16(a) shows when sunshine reflections create haziness around object with distance of over18.3M and cloudy scene Figure 16(b) with small object at far distance of over 27 meters. All these detections were made using YOLOv5s which is selected based on use case based on FPS cap versus performance trade off.

The MFLD model using YOLOv5s is integrated into the ESA module and provides the ESA message through the Vision Integration protocol (VIP). This ESA message is utilized to enhance algorithms and modules in accordance with the work done in MOOS-IvP. The USV was deployed to navigate an obstacle position, and in place of using LiDAR for object detection, depth camera was employed, which achieved a fair success. Notably, the USV detected all pre-identified objects while minimizing the occurrence of false positives by the depth camera with bounding boxes around the detected objects using improvised detection and ESA module. This detection approach was critical in enhancing the safety of the navigation system for short-term obstacle avoidance. The test environment used in this study simulated detected objects by generating uniformly random object points within selected small *xy* regions of the operational area with external sensor data in pMarineViewer.

B. FIELD EXPERIMENT

Extensive field experiments were conducted under multiple scenarios to evaluate the practical feasibility of the developed autonomous navigation system utilizing the Suraya Surveyor using proposed ESA. The experiments involved different objects, as demonstrated in Figure 17, to show the capabilities of the proposed approach in typical maritime environments.

We developed an intuitive interface for operators to modify autonomy algorithms and command autonomous navigation from a ground station. Object detection and tracking were implemented in Python, while MOOS-IvP was coded in C++. This study reports results from three field experiments

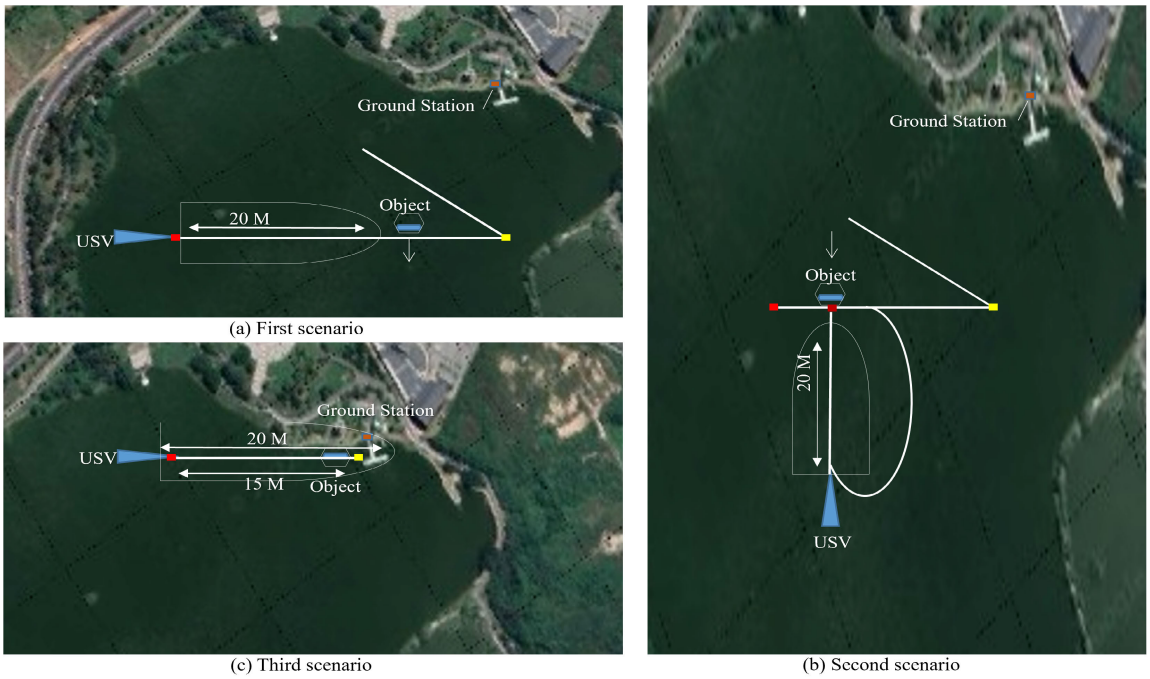


FIGURE 18. Illustrates the experimental scenarios for multiple objects tracking and collision avoidance. The path planning and directions of the objects are depicted. The alert range is visualized as a shaded area, and a hexagon-shaped buffer zone is introduced.

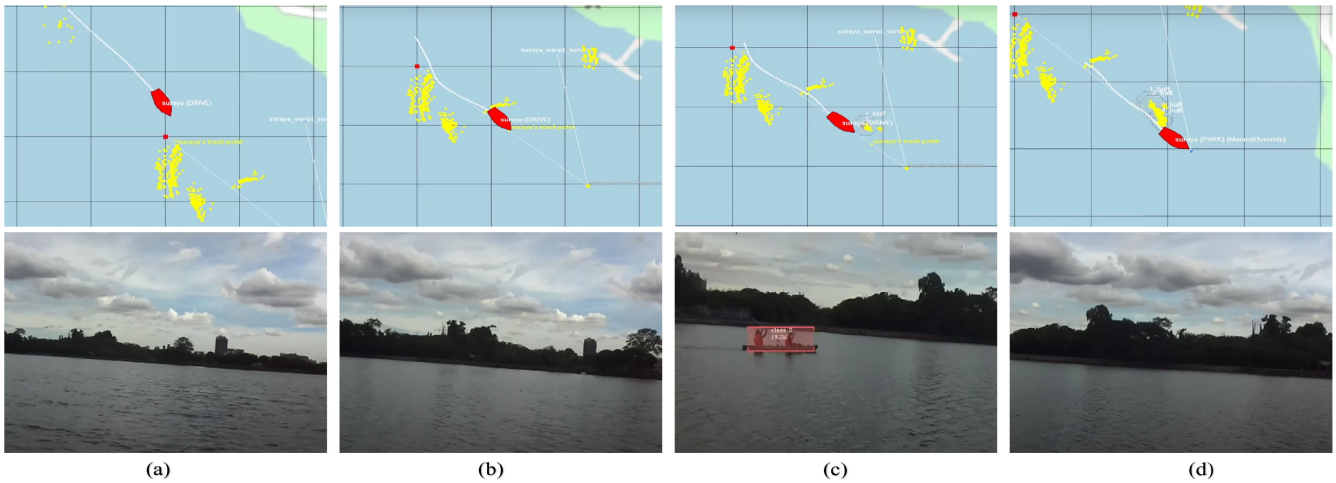
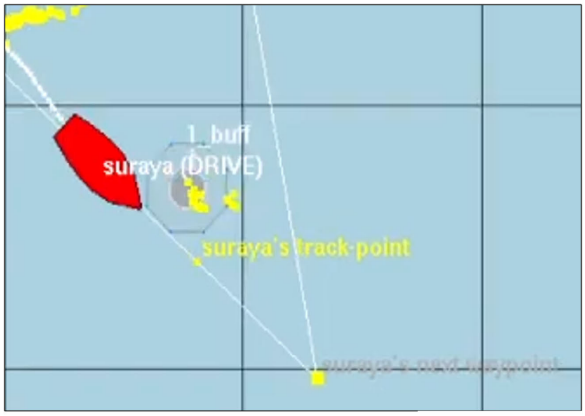


FIGURE 19. Trajectories of Suraya USV during the field experiment of the first scenario. On the upper side is the USV trajectory and the generated Buffer Obstacle on pMarineViewer. The figures on the lower side show corresponding snapshots captured by ZED2i from the Virtual Network Computing (VNC) computer on the ground station for object detection with the bounding boxes.

during Suraya Surveyor developments. Figure 18 shows test scenarios in the Kepong Metropolitan Lake Garden with two distinct obstacles. Operators manually controlled the objects while the custom detector provided reliable information to ensure USV safety.

In the first scenario showed in Figure 19, the placement of obstacles near the path of the USV allowed ZED2i to detect them without posing a risk of collision. In the second scenario, the USV had to retrace its course to reach a new starting point, while en route, the USV encountered an object moving in the opposite direction. The third scenario involved the USV traveling ahead and encountering a stationary obstacle along its path.

The first scenario depicts the USV and object following trajectories that intersect, but the object remains outside the designated safe zone while the USV continues towards its waypoint. This should result in no deviation of the USV from its planned path. In this scenario, obstacles were placed next to the path of the USV in such a way as to allow for their detection by ZED2i without posing a collision risk.



(a)

(b)

FIGURE 20. Zoomed area of obstacle encounter and buffer generation. (a) generated Buffer Obstacle on pMarineViewer. (b) The detected object snapshots captured by ZED2i on VNC.

The USV initiated on its trail, traveling at a speed of 2 knots, along the solid line path which represents the planned sequence of waypoints. This path runs parallel to the dotted line in the image, which indicates the USV's trajectory by the onboard navigation sensors utilizing updated GPS, compass, and INS data. In Figure 19 (a), the USV's autonomous mode successfully locks onto its starting point (red square) using navigational data. Figure 19 (b) demonstrates the USV's navigation to a waypoint without encountering any obstacles. Figure 19 (c) illustrates the first encounter with an obstacle within the line of sight and alert range. At this point, MOOS-IvP receives ESA messages and generates the buffer obstacle. In Figure 19 (d), after the USV crosses the obstacle encounter zone ensuring that the USV remains on its intended path without any deviation. Figure 20 illustrates the zoomed area of obstacle encounter and buffer generation, showcasing the generated Buffer Obstacle on pMarineViewer, along with the captured snapshots of the detected objects by ZED2i on VNC.

The second scenario focused on testing the USV capabilities in a more complex situation. Figure 21(a) illustrates when the USV firstly operated in autonomous mode, following a predetermined path. However, it encountered a situation where it needed to establish a new starting point for its trajectory. This new starting point was located along the pre-defined path, requiring the USV to retrace its course to reach it. While en route to the new starting point in Figure 21(b), the USV encountered an object sailing in the opposite direction, necessitating deviation action. The controller received pertinent ESA messages and promptly generated a buffer obstacle to facilitate collision avoidance, shown in Figure 21(c). In Figure 21(d) the system successfully circumvented the obstacle and resumed its intended path as per the pre-planned trajectory.

In contrast to the previous scenarios, the third scenario shaped at a speed of 2 knots (1.03 m/s) on a short and planned path of 15 meters within the alert range. As a result, the USV's path was blocked from the port side, and the system employed a collision avoidance behavior that remained almost constant throughout the scenario, ESA initiated immediately upon entering the planned path throughout the trail. However, when the USV was within the safe range behind a pier, the desired behaviour was to be directed on the starboard side to avoid objects. Regrettably, the system moved towards the portside, as shown in Figure 22, which was undesirable. Ideally, the USV should have continued to travel toward the starboard side. This demonstrates the robustness of the ESA method, establishing it as a fundamental component for ensuring safe navigation. However, it is essential to couple it with well-trained MOOS-IvP behaviours to ensure safe operations.

The analysis of Table 14, featuring a 73-frame VNC video, reveals significant disparities between the expected and observed object behaviors, which raise concerns about the accuracy of MOOS-IvP behaviors. We display the initial and final five outputs to address the comprehensive result length. This demonstrate that the communication system responsible for transmitting data via UDP to the controller performed flawlessly without losing any detections or affecting the performance of the custom detector. Nevertheless, the detector continued to provide object data to the communication system, which transmitted it to the controller without any issues.

TABLE 14. Third scenario observations of expected and observed behaviours.

Frame number	Detected	Expected behaviour	Observed behaviour
1	✓	Starboard-directed	Moved to portside
2	✓	Starboard-directed	Moved to portside
3	✓	Starboard-directed	Moved to portside
4	✓	Starboard-directed	Moved to portside
5	✓	Starboard-directed	Moved to portside
69	✓	Stay on course	Moved to portside
70	✓	Stay on course	Moved to portside
71	✓	Stay on course	Moved to portside
72	×	Stay on course	Moved to portside
73	×	Stop	Manually stopped

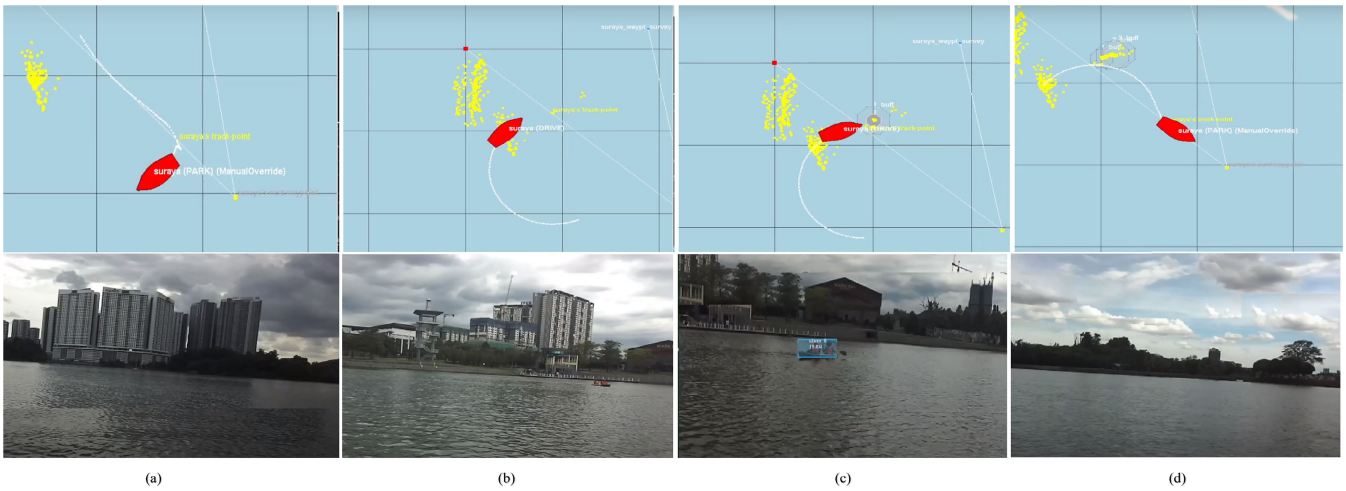


FIGURE 21. pMarineViewer screenshot of the second scenario and ZED2i detection.

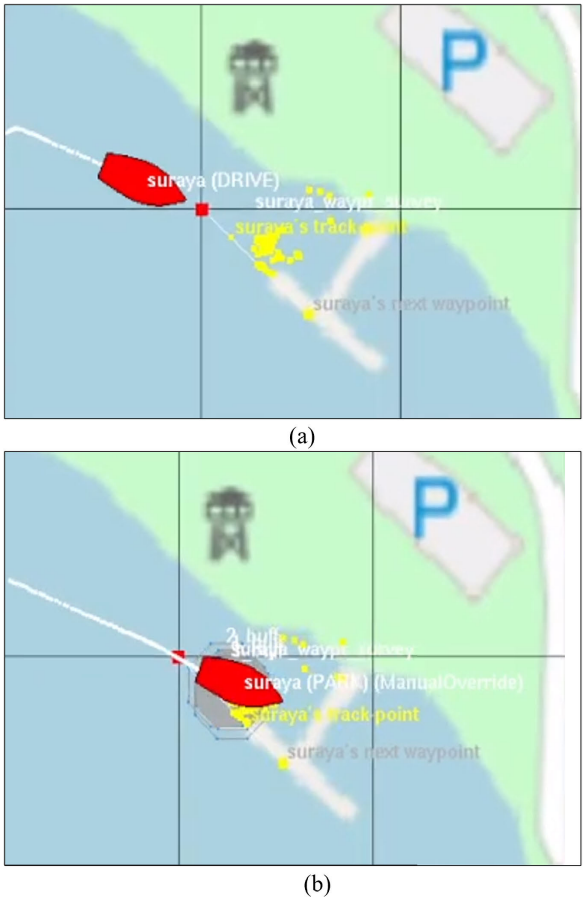


FIGURE 22. USV's path and obstacle avoidance behaviour of the third scenario in pMarineViewer.

Table 15 provides detailed information about the NMEA sentence from proposed ESA module transmitted via UDP from the custom detector to the controller, including the number of objects detected in the frame, Object ID, and Checksum value, all presented in HEX. Specifically, the table shows the NMEA sentence information for the third scenario when the boat entered the planned path and detected objects in the first frame itself

TABLE 15. Third scenario data transmission samples.

No	NMAE Format
1	$ODOBJ,1,1,110,1.429,0.924,6010.742,0.506,1.217,0.506,15.812*72
2	$ODOBJ,2,1,110,1.467,0.892,5336.914,0.495,1.148,0.495,15.559*75
3	$ODOBJ,2,2,109,0.024,0.353,4692.383,4.076,1.792,4.076,16.227*55
4	$ODOBJ,2,1,110,1.837,0.268,8725.586,0.468,1.369,0.468,4.671*6A
5	$ODOBJ,2,2,109,2.661,0.259,6328.125,0.955,0.406,0.955,5.404*69
69	$ODOBJ,2,1,110,1.828,0.265,7924.805,0.475,1.353,0.475,4.538*68
70	$ODOBJ,2,2,109,2.633,0.258,6328.125,0.887,0.405,0.887,5.301*6E
71	$ODOBJ,2,1,110,1.828,0.257,8950.195,0.474,1.340,0.474,4.401*6C
72	$ODOBJ,2,2,109,2.602,0.274,6328.125,0.817,0.405,0.817,5.198*60
73	None

An obstacle alert is posted as the object comes within the alert range. This posting will spawn a new obstacle avoidance behaviour via the updated parameter discussed in obstacle avoidance manager subsection. In this field test a typical posting looks like the following example, which was taken during the third scenario:

OBSTACLE_ALERT "name=3#poly=pts={46.07, 16.67:48.24,14.5:48.24,11.44:46.07,9.27:43.01, 9.27:40.84,11.44:40.84,14.5:43.01,16.67}, label=3"

The first time this alert variable is published, it can be regarded as an alert because the obstacle's existence is new information for whoever subscribes to these alerts (typically the helm). Moreover, an example is shown below for the form of TRACKED_FEATURE of MOOS variable, which is subscribed for the obstacle manager during the third scenario:

TRACKED_FEATURE "x=23.2,y=19.8, label=1,color=1,size=2"

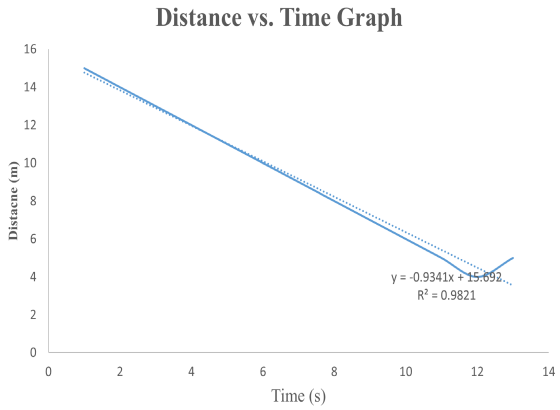


FIGURE 23. Distance between USV and Objects in Third Field Test Scenario.

The viewpoint is a particular message that the pMarineViewer application knows how to parse and render. For this case, a typical posting during the third scenario: looks like this:

VIEW_POINT "x=45.94,y=28.4,active=true, label=166438610108,msg=3, type=obstacle,label_color=invisible, vertex_color=yellow, vertex_size=5"

Figure 23 displays the distance between the USV, and the object encountered in the third scenario of the field test. The Y-axis represents the distance in meters, while the X-axis represents time in seconds. As depicted in this scenario, the distance from the USV to the objects decreased gradually as the USV travelled on the planned path of 15 meters, with the ESA detecting an obstacle upon entering the path of the set waypoint within the alert range. The graph shows that the distance from the USV to the objects decreased to its minimum value of 4 meters near zone buffer before deviating and moving away to 5 meters, indicating the successful implementation of the system performance for collision avoidance. However, collision avoidance behaviour initiated is not desirable and MOOS-IvP Controller need further attention.

VII. DISCUSSION

The Suraya USV is designed for autonomous navigation with appropriate hardware, software algorithms, and optimum integration. We show how the effective use of stereo vision enhances situational awareness and makes USV fully autonomous. Our system facilitates smooth data transfer and seamless integration of the computing model and navigation sensors, eliminating the need for additional mechanical design. To the best of our knowledge, this approach has not been undertaken in a manner as described herein. The system enables well-informed navigation decisions by real-time fusion of data from the stereo vision system and navigational data. This novel integration approach sets it apart from existing systems. Using dual thrusters, a vision integration protocol and an AI computer with a computing model simplify the system, improving existing solutions. Furthermore, our integration of MOOS-IvP on the USV shows this framework could still be a standout choice for maritime operations, making MOOS-IvP implementation easier on marine platforms. This paves the way for future studies to explore more straightforward integration methods and cost-effective sensor options.

The software module employs stereo cameras, specifically ZED2i, to enhance situational awareness through obstacle detection. It also utilizes YOLOv5, and MOOS-lvP frameworks to achieve accurate object detection and collision avoidance. The architectural design enables precise obstacle detection, efficient communication, and intuitive control, significantly enhancing situational awareness in USVs. Vision Integration Protocol (VIP) facilitates the selection of the optimal MOOS behaviour for streamlined communication.

The vision integration protocol ensures reliable communication and surveillance using UDP, thereby assisting in object tracking, collision avoidance, and providing crucial pixel positions and 3D information. This protocol operates effectively on the Jetson Xavier AGX platform, enabling seamless interaction with navigation sensors. By transmitting object detection and tracking data in NMEA format via UDP, efficient control of the USV is facilitated. The proposed VIP enhances operational prospects and ensures reliable USV operations.

The results obtained from the collision avoidance experiments showed that the outcomes were influenced by the size and type of obstacles. It was recommended that the estimated area of an obstacle should be one-third of the detection range when the USV is moving at a speed of 2 knots. Different measurements were obtained for different obstacles, and MOOS-lvP dynamically adjusted the buffer obstacle's size as the USV approached it. The yellow dots depicted in the scenario figures provided crucial sensor information, updated approximately every 0.05 seconds. MOOS-lvP prioritized collision avoidance behaviour and relied on accurate obstacle buffer data to ensure safe navigation. Overall, Suraya USV successfully navigated through complex scenarios, demonstrating the effectiveness of its collision system.

In the three presented scenarios, the ESA promptly notified the navigation system within an appropriate range. The USV successfully avoided collisions in the first two scenarios with ample time, but encountered challenges in the third scenario where obstacles were closer to the vessel than the designated safe path. To improve performance, advanced decision-making strategies considering obstacle classification and dynamic capabilities are necessary. In summary, our work resulted in a robust object detection model for secure USV navigation, adept at detecting short- to mid-range obstacles. The implementation of a software module using a stereo camera was able to achieve Enhance Situational Awareness (ESA). Also, the integration protocol

ensures seamless utilization of object detection and collision avoidance through MOOS-IvP. A comprehensive field test validated the system's performance, affirming its reliability in practical maritime applications.

Further experimentation, research, and development are warranted to gather substantial statistical data on the safety of autonomous vessels. This entails testing a wider range of scenarios in uncontrolled environments. Additionally, implementing dynamic fault-tolerant strategies would enhance the behaviour and prevent detrimental decisions by the USV. Investigation of the system's performance under varying weather conditions and the incorporation of an infrared sensor for nighttime operations can further enhance situational awareness. The fusion of active sensors would provide valuable data to augment long-distance vision capabilities.

VIII. CONCLUSION

This study presented the successful development of an autonomous navigation-enabled USV system and presents the corresponding field test results. The system encompasses the hardware platform of the vehicle, operational software algorithms, and the integration of hardware and software components tailored specifically for USV applications. Throughout the development process, our focus was on leveraging stereo vision technology to enhance situational awareness and implementing a collision avoidance system to ensure safe USV operations. The field experiments conducted in real environments effectively demonstrated the performance and practical feasibility of the developed USV and autonomous navigation system. Moreover, we have provided compelling evidence from challenging scenarios that validate our approach's capability to furnish the controller with reliable information regarding surrounding obstacles, thanks to our customized detector module built with YOLOv5 and ZED2i. Furthermore, the results obtained from the vision system integration protocol substantiate the impeccable transmission of data without any loss or adverse impact on the performance of the custom detector. It is important to note, however, that the collision avoidance results are insufficient to provide comprehensive safety assurance, as evidenced in the third scenario in our field test experiment. Consequently, these observations motivate the need for further work on robust detection, safe navigation, and the imperative for continued research to enhance the safety aspects of USV navigation.

REFERENCES

[1] L. Duan, B. Luo, Q.-Y. Li, and G.-h. Yu, "Research on intelligence, surveillance and reconnaissance mission planning model and method for naval fleet," in *Proc. Chin. Control Decis. Conf. (CCDC)*, May 2016, pp. 2419–2424.

[2] W. Naeem, R. Sutton, and J. Chudley, "Modelling and control of an unmanned surface vehicle for environmental monitoring," in *Proc. UKACC Int. Control Conf.*, 2006, pp. 1–6.

[3] M. Specht, C. Specht, H. Lasota, and P. Cywiński, "Assessment of the steering precision of a hydrographic unmanned surface vessel (USV) along sounding profiles using a low-cost multi-global navigation satellite system (GNSS) receiver supported autopilot," *Sensors*, vol. 19, no. 18, p. 3939, Sep. 2019.

[4] J. Han, Y. Cho, J. Kim, J. Kim, N. Son, and S. Y. Kim, "Autonomous collision detection and avoidance for ARAGON USV: Development and field tests," *J. Field Robot.*, vol. 37, no. 6, pp. 987–1002, Sep. 2020.

[5] P. Robinette, M. Sacarny, M. DeFilippo, M. Novitzky, and M. R. Benjamin, "Sensor evaluation for autonomous surface vehicles in inland waterways," in *Proc. OCEANS Marseille*, Jun. 2019, pp. 1–8.

[6] B. Cole, M. R. Benjamin, and S. Randeni, "AIS-based collision avoidance in MOOS-IvP using a geodetic unscented Kalman filter," in *Proc. OCEANS San Diego Porto*, Sep. 2021, pp. 1–10.

[7] J. Muhovic, R. Mandeljc, B. Bovcon, M. Kristan, and J. Perš, "Obstacle tracking for unmanned surface vessels using 3-D point cloud," *IEEE J. Ocean. Eng.*, vol. 45, no. 3, pp. 786–798, Jul. 2020.

[8] B.-S. Shin, X. Mou, W. Mou, and H. Wang, "Vision-based navigation of an unmanned surface vehicle with object detection and tracking abilities," *Mach. Vis. Appl.*, vol. 29, no. 1, pp. 95–112, Jan. 2018.

[9] M. R. Benjamin, H. Schmidt, P. M. Newman, and J. J. Leonard, "Nested autonomy for unmanned marine vehicles with moos-ivp," *J. Field Robot.*, vol. 27, no. 6, pp. 834–875, 2010.

[10] *Hidrokinetik*. Accessed: Jun. 15, 2023. [Online]. Available: https://hidrokinetik.com/

[11] J. Park, J. Kim, and N. Son, "Passive target tracking of marine traffic ships using onboard monocular camera for unmanned surface vessel," *Electron. Lett.*, vol. 51, no. 13, pp. 987–989, Jun. 2015.

[12] A. Gonzalez-Garcia, I. Collado, R. Cuan-Urquizo, R. Reyes, and L. Garrido, "A 3D vision based obstacle avoidance methodology for unmanned surface vehicles," in *Proc. 21st Congreso Mexicano de Robótica*, 2019, pp. 1–7.

[13] R. Fan, L. Wang, M. J. Bocus, and I. Pitas, "Computer stereo vision for autonomous driving," 2020, *arXiv:2012.03194*.

[14] T. Huntsberger, H. Aghazarian, A. Howard, and D. C. Trotz, "Stereo vision–based navigation for autonomous surface vessels," *J. Field Robot.*, vol. 28, no. 1, pp. 3–18, Jan. 2011.

[15] B. Bovcon, R. Mandeljc, J. Perš, and M. Kristan, "Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation," *Robot. Auto. Syst.*, vol. 104, pp. 1–13, Jun. 2018.

[16] M. Kristan, V. Sulic Kenk, S. Kovacic, and J. Perš, "Fast image-based obstacle detection from unmanned surface vehicles," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 641–654, Mar. 2016.

[17] ZED Product Portfolio, *Stereolabs Product Portfolio and Specifications; Revision 1*, Stereolabs, Orsay, France, 2022.

[18] L. E. Ortiz, V. E. Cabrera, and L. M. G. Goncalves, "Depth data error modeling of the ZED 3D vision sensor from stereolabs," *ELCVIA Electron. Lett. Comput. Vis. Image Anal.*, vol. 17, no. 1, pp. 1–15, Jun. 2018.

[19] J. Zhang, "Perception and motion planning for autonomous surface vehicles in aquaculture," Ph.D. dissertation, Dept. Elect. Eng. Comput., Massachusetts Inst. Technol., Cambridge, MA, USA, 2022.

[20] J. Park, Y. Cho, B. Yoo, and J. Kim, "Autonomous collision avoidance for unmanned surface ships using onboard monocular vision," in *Proc. OCEANS MTS/IEEE Washington*, Oct. 2015, pp. 1–6.

[21] W. Liu, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.

[22] S. Alashhab, A.-J. Gallego, A. Pertusa, and P. Gil, "Precise ship location with CNN filter selection from optical aerial images," *IEEE Access*, vol. 7, pp. 96567–96582, 2019.

[23] 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)*, Jun. 2016, pp. 779–788.

[24] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimedia Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023.

[25] T. Liu, B. Pang, L. Zhang, W. Yang, and X. Sun, "Sea surface object detection algorithm based on YOLO v4 fused with reverse depthwise separable convolution (RDSC) for USV," *J. Mar. Sci. Eng.*, vol. 9, no. 7, p. 753, Jul. 2021.

[26] G. Chen, J. Qi, and Z. Dai, "Real-time maritime obstacle detection based on YOLOv5 for autonomous berthing," in *Proc. Int. Conf. Bio-Inspired Comput., Theories Appl. (BIC-TA)*, Taiyuan, China, Dec. 2021, pp. 412–427.

[27] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "SeaShips: A large-scale precisely annotated dataset for ship detection," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2593–2604, Oct. 2018.

[28] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 1993–2016, Aug. 2017.
[29] P. Tang, R. Zhang, D. Liu, L. Huang, G. Liu, and T. Deng, "Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle," *Ocean Eng.*, vol. 106, pp. 128–140, Sep. 2015.
[30] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "MPC-based collision avoidance strategy for existing marine vessel guidance systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7618–7623.
[31] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with COLREGS, using velocity obstacles," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 110–119, Jan. 2014.
[32] M. Hepworth, V. Garofano, Y. Pang, and V. Reppa, "A stereovision-based navigation system for autonomous inland vessels," in *Proc. Int. Ship Control Syst. Symp.*, vol. 16, 2022, p. 6.
[33] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. RA-2, no. 1, pp. 14–23, Mar. 1986.
[34] M. R. Benjamin, *Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control*. Providence, Rhode Island: Brown Univ., 2002.
[35] StereoLabs. (2023). *ZED 2I Camera*. [Online]. Available: https://www.stereolabs.com/zed-2i/
[36] Canal Geomatics. (2023). *KVH C100 Compass Engine*. [Online]. Available: https://www.canalgeomatics.com/product/kvh-c100-compass-engine/
[37] SBG. (2022). *SBG Elipse-E*. [Online]. Available: https://www.sbg-systems.com/products/ellipse-series/
[38] *MBR-179 Product Documentation*. Accessed: Jun. 15, 2023. [Online]. Available: https://www.kongsberg.com/globalassets/maritime/km-products/product-documents/mbr-179.pdf
[39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.
[40] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, J. Fang, K. Michael, D. Montes, J. Nadar, P. Skalski, *Ultralytics/YOLOv5: V6. 1-TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference*. Zenodo, 2022.
[41] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Heidelberg, Germany: Springer, 2014, pp. 740–755.
[42] J. Becktor, F. E. T. Schöller, E. Boukas, M. Blanke, and L. Nalpantidis, "Bolstering maritime object detection with synthetic data," *IFAC-PapersOnLine*, vol. 55, no. 31, pp. 64–69, 2022.
[43] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2022, pp. 925–944.
[44] M. R. Benjamin, M. Defilippo, P. Robinette, and M. Novitzky, "Obstacle avoidance using multiobjective optimization and a dynamic obstacle manager," *IEEE J. Ocean. Eng.*, vol. 44, no. 2, pp. 331–342, Apr. 2019.
[45] *pObstacleMgr*. Accessed: Jun. 15, 2023. [Online]. Available: https://oceanai.mit.edu/ivpman/apps/pObstacleMgr
[46] *PNodeReporter*. Accessed: Jun. 15, 2023. [Online]. Available: https://oceanai.mit.edu/ivpman/pmwiki/pmwiki.php?n=IvPTools.PNodeReporter
[47] *PMViewer*. [Online]. Available: https://oceanai.mit.edu/ivpman/pmwiki/pmwiki.php?n=IvPTools.PMViewer
[48] *UProcessWatch*. Accessed: Jun. 15, 2023. [Online]. Available: https://oceanai.mit.edu/moos-ivp/pmwiki/pmwiki.php?n=Manifest.UProcessWatch

YOUSEF ABD ALHATTAB received the master's and Ph.D. degrees in mechatronics engineering from International Islamic University Malaysia (IIUM). He is currently an ML Consultant and a Research Fellow with the Centre for Unmanned Technologies (CUTe). His research interests include signal processing, object detection, and unmanned surface vessels (USV). He has extensive experience in deep learning models for glass break detection, urban sound classification, and maritime object detection. His ongoing Ph.D. research focuses on improving and developing a fusion tool for obstacle detection and collision avoidance on USV using AI devices.

ZULKIFLI BIN ZAINAL ABIDIN received the B.Eng. degree in computer and information engineering from International Islamic University Malaysia (IIUM), in 2003, and the M.Sc. and Ph.D. degrees from USM, in 2007 and 2013, respectively. He is currently the Director and the Co-Founder of the Center for Unmanned Technologies (CUTe) and he has secured research projects and grants exceeding RM5 million. He is also a Consultant and a Subject Expert for various industries and government agencies, including PDRM, PROTON, MARA, and Petronas. He is a Roboticist. He is a Chartered Engineer (C.Eng) and actively contributes to international and local publications and conferences. He is a member of the SMIEEE Oceanic Engineering Society, Malaysia, where he is also the Vice Chairperson 90250137. His research interests include underwater, ground, and space robotic platform development, sensing devices, and intelligent control algorithms. He is also a Trainer in off-road motorsport.

AHMED RIMAZ FAIZABADI (Student Member, IEEE) is currently pursuing the Ph.D. degree in computer vision and deep learning with the Department of Mechatronics, International Islamic University Malaysia (IIUM), Malaysia. He is also a Researcher with over 15 years of experience in research and development, teaching, and industry. He was an Invitee with the SRI International Summer School for Formal Methods. He was also with the General Motors Research and Development India Science Laboratory (GM-ISL) and other industries of repute. He has published over a dozen peer-reviewed research articles and authored two books. He is a Student Member of professional societies SAE-M. He has volunteered as the President of PGSA-KoE IIUM (2019–2022), the Head (Education and Research) PGSS IIUM (2020–2021), and a Treasurer, IEEE SB IIUM (2021–2022). Society of Automotive Engineers Malaysia (SAE-M) has awarded him an Outstanding Student of 2021.

HASAN F. M. ZAKI received the Ph.D. degree in computer science from The University of Western Australia. He is currently an Assistant Professor in mechatronics engineering with International Islamic University Malaysia and the Head of Embedded AI with the Centre for Unmanned Technologies (CUTe). His research interests include robotic vision, RGB-depth object and scene recognition, machine learning, 3D face analysis, and action recognition.

AHMAD IMRAN IBRAHIM (Member, IEEE) received the Ph.D. degree in mechanical engineering from the University of Bristol, U.K. He is currently an Assistant Professor in mechatronics engineering with International Islamic University Malaysia and the Head of Marine Robotics with the Centre for Unmanned Technologies (CUTe). His research interests include robotic design, underactuated systems, and robotics optimization.