IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 20 December 2023, accepted 30 December 2023, date of publication 9 January 2024,
date of current version 17 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3351764

RESEARCH ARTICLE

Towards Continuous Floating Invasive Plant Removal Using Unmanned Surface Vehicles and Computer Vision

EDWIN SALCEDO, YAMIL UCHANI, MISAEL MAMANI, (Student Member, IEEE), AND MARIEL FERNANDEZ
Centro de Investigación, Desarrollo e Innovación en Ingeniería Mecatrónica, Universidad Católica Boliviana "San Pablo," La Paz, Bolivia
Corresponding author: Edwin Salcedo (esalcedo@ucb.edu.bo)

ABSTRACT Deficient domestic wastewater management, industrial waste, and floating debris are some leading factors that contribute to inland water pollution. The surplus of minerals and nutrients in overly contaminated zones can lead to the invasion of different invasive weeds. *Lemnoideae*, commonly known as duckweed, is a family of floating plants that has no leaves or stems and forms dense colonies with a fast growth rate. If not controlled, duckweed establishes a green layer on the surface and depletes fish and other organisms of oxygen and sunlight. In the last decades, duckweed blooms have become an expensive plague to remove in Lake Titicaca, causing significant biodiversity and economic losses. To address this issue, we propose the development of an unmanned surface vehicle (USV) to continuously monitor and collect duckweed. This device includes a steering angle estimation system (SAE) that combines a duckweed detector, a steering angle classifier, and a momentum-inspired module to determine continuously the orientation of the USV when it is inspecting a zone. We train and test our deep learning-based models with two novel datasets: first, a dataset for end-to-end steering angle classification collected synthetically using a bespoke simulator; and second, a dataset for duckweed detection. An experimental evaluation of the USV and SAE system is provided in virtual and outdoor environments.

INDEX TERMS Duckweed detection, unmanned surface vehicle, steering angle estimation, virtual to real learning.

I. INTRODUCTION

Aquatic weeds are plants that are not native to a region and that spread through interconnected, still water bodies by waterfowl, livestock, boat trailers, and fishermen's gear and boots. Although they can be beneficial in providing natural filtration, food, and shelter for aquatic animals, they can become an aggressive plague when they grow out of control. Moreover, while they are manageable in small water bodies, including ponds, pools, and wetlands, their removal is challenging in longer water bodies, such as lagoons, lakes, dams, and bays. They disrupt such ecosystems by depleting light and oxygen resources, and over-increasing minerals and nutrients, to the point of causing fish death and sludge [1]. This process, known as eutrophication, causes ecosystem decay and biodiversity loss that are hard to recover in years.

According to McNabb and Anderson [2], aquatic weeds can be clustered into two groups: algae and vascular plants. Algae are characterized mainly by being a common component and critical foundation of the food chain in healthy aquatic environments. A rapid increase in the population of uni-cellular algae, also known as harmful algae blooms (HABs), has been a constant research thread for automatic removal systems due to their dangerous consequences to nature [3], [4]. On the other hand, vascular plants can be split into three sub-categories: submerged, emergent and

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

floating weeds. The first two are rooted to the bottom and can remain underwater or appear over the surface, respectively. In contrast, floating invasive weeds grow freely over the water surface, and they find suitable conditions to overgrow in eutrophicated water bodies that have a surplus of minerals and nutrients, an increase in temperature, and little airflow. The development of new automatic tools is important to control and suppress invasive weeds timely.

The present research was developed as a response to the recent duckweed blooms in Lake Titicaca. Duckweed, also known as (*lemna minor*), is a species of floating weed in the subfamily *lemnoideae*. In recent decades, increasing urbanization in coastal areas, inadequate sewage, and domestic wastes have poured a surplus of minerals and nutrients [5] in Lake Titicaca, affecting the water consumption and sustenance of 3 million residents in Bolivia and Peru. Furthermore, since 2015, the effects of eutrophication and climate change have worsened, and duckweed blooms have become more frequent [6]. Nowadays, duckweed invasion is a recurring problem in Cohana Bay, Bolivia and Puno Bay, Peru, where the local governments are frequently forced to spend large amounts of money to treat and dredge the lake.

Compared with other elements present in aquatic environments (e.g. boats and buoys), duckweed colonies can be arbitrarily shaped, and their precise detection is a challenging task; see Figure 1. In this research, we propose the development of a low-cost unmanned surface vehicle (USV) for the monitoring and removal of duckweed from Lake Titicaca. With this goal, we implement and deploy a Steering Angle Estimation system (SAE) to process RGB and stereo vision data coming from an OAK-D camera in real-time. Not only does the SAE system recognize duckweed appearances, but it also avoids collisions when the USV keeps cleaning a water body. The gradual development of the SAE's architecture and subsequent deployment on the embedded system NVIDIA Jetson Nano Development Kit makes the autonomous navigation approach stable and portable. Moreover, a virtual simulator was implemented for synthetic data collection and experimentation of the removal system. The effectiveness of the duckweed detection models, SAE system, and removal device were tested reaching high precision results. To summarize, the contributions of this paper include:

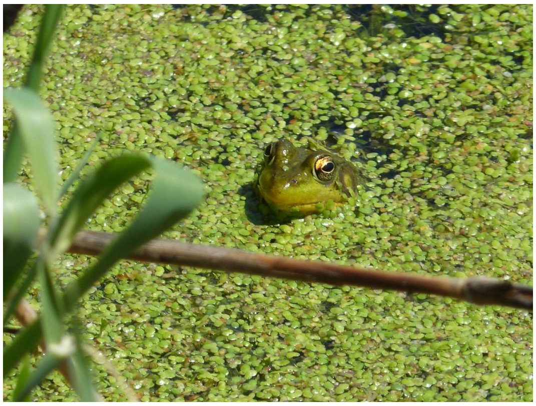


FIGURE 1. Complex shape of a duckweed colony covering the entire water surface and not supporting a precise detection. Courtesy of GBIF [7].

- A survey on the recent automatic tools and robotic-based approaches for aquatic weed detection and removal.
- The prototype of the electronic and mechanical sub-systems of an unmanned duckweed collection vehicle for experimental evaluation.
- The development of duckweed detectors and steering angle estimators.
- The implementation of a virtual simulator to augment data and evaluate the autonomous removal system.

The rest of this paper is organized as follows. In Section II, we review recent approaches for aquatic weed mapping and removal that make use of different sensing and collection modalities, but we pay particular attention to unmanned surface vessels that deploy modern artificial intelligence systems. In Section III, we present the development stages of the proposed unmanned surface vehicle. In Section IV, we describe the implementation of the autonomous navigation system. Afterwards, Section V informs about the obtained results when testing the removal system in situ and inside a simulation environment. The paper ends with a conclusion section in VI.

II. LITERATURE REVIEW

In this section, a review of automatic methods and unmanned vehicles to collect and monitor invasive weeds is discussed. In addition, we include an analysis of the related work on self-driving approaches that could be a good basis for the development of the proposed aquatic vehicle.

A. FLOATING WEED REMOVAL METHODS

Nowadays, there are different hand-operated methods to prevent and control invasive weed growth in small water bodies, including ponds, pools, and wetlands. For instance, handheld tools, such as rakes or planks covered with fabric to push weed to the edges; chemical solutions, such as *diquat, metsulfuron methyl*, *chlorimuron methyl*, and *fluridone* [14], [15]; and biological control, such as koi, goldfish, and grass carp. Although the aforementioned methods proved to be efficient for small ponds, they can be impractical in larger water bodies. Not only does their application require support from expert professionals, but also a thorough study of the potential side effects in the long term. For example, authors in [16] explain how efforts to remove invasive plants by introducing Asian species of carp caused the loss of native fish populations in the Mississippi River system.

When it comes to mechanical means, private companies in the UK and the US, such as Elastec [17], Oilskimmers Inc. [18], and Vikoma [19], offer their own models of aquatic weed skimmers. These are mainly based on suction pumps and filters that drain water to land and then to the source while filtering duckweed. Other recommended methods are based

TABLE 1. Summary of recent unmanned vehicles proposed for invasive weed removal and mapping.

Year & Reference	Vehicle type/design	Autonomy Approach	Target plant species	Sensing modalities	Detection Model/ Method	End device	Main Task	Operation Time	Weed Type
2022 [8]	USV (catamaran), UAV (quadrotor)	Mission Planner	Hydrilla verticillata	Hydroacoustic imaging, UAV camera	DenseNet, ResNet, Sensor Fusion	Pixhawk Autopilot	Herbicide dispersal	2 Hrs	SAV
2021 [9]	USV (trimaran)	ROS-based software	Eurasian water-milfoil	SONAR, RGB camera	None	NVIDIA Jetson Nano	Data collection	Not specified	SAV
2021 [10]	None	None	General	RGB camera	K-Means	Raspberry Pi 3B	Weed detection	Not specified	Floating Plants
2019 [11]	USV (catamaran)	Mission Planner	Hydrilla verticillata	Hydroacoustic imaging	Alexnet	Pixhawk Autopilot	Herbicide dispersal	2 Hrs	SAV
2018 [12]	UAV (quadrotor)	None	Ribbonweed, Pondweed, Parrot's feather, Cumbungi	UAV spectral camera, remote sensing	Google Earth Enginee Toolkit	None	Mapping	None	Floating Plants
2017 [4]	USV (catamaran), UAV (quadrotor)	GNC system, LoS	Harmful algae bloom	UAV camera	LBP	Intel NUC	Algae removal	2.5 Hrs	Algae
2010 [13]	UAV (single rotor)	COTS software	Salvinia	UAV camera	SVMs	COTS autopilot engine	Mapping	Not specified	Floating Plants

on bubble aerators whose continuous operation can avoid aquatic weed overgrowth. The main disadvantage of these methods is that they would require multiple devices and high energy costs when applied to larger water bodies. Removing floating invasive plants from lagoons, lakes, dams, and bays usually requires heavy machinery (e.g. backhoes), which in turn incurs high costs.

B. UNMANNED VEHICLES FOR AQUATIC WEED AND ALGAE MANAGEMENT

During the past decades, USVs have demonstrated tremendous potential to automate tasks performed in aquatic environments, such as navigation, sensor data collection, and debris removal. In contrast, the body of knowledge about autonomous vehicles for floating weed removal is still narrow. To the best of the authors' knowledge, USVs listed in Table 1 are the most recent proposals, yet their target are different aquatic weeds. These range from submerged aquatic vegetation, such as *hydrilla*, *eurasian watermilfoil*, and *contail*; emergent plants, such as *alligatorweed* and *water primrose*; and floating weeds, such as *water meal*, *water hyacinth*, and *sargassum*. For instance, Patel et al. propose the development of a two-hull unmanned boat that uses hydroacoustic imaging to map regions with *hydrilla* and disperse *aquathol-K* for their removal [11]. Additionally, S. Jung et al. designed a catamaran-shape USV to carry a electrocoagulation and flotation reactor to separate pollutants and cyanobacteria from superficial water by using a series of parallel electrodes inside a box to generate an electrolytic reaction [4]. While the left electrodes (anodes) flocculate the microalgae particles, the hydrogen bubbles generated from the right electrodes (cathodes) float the flocculated mass, creating an effective mechanism to separate and collect HABs from lakes.

Most USVs proposed for aquatic weed management aim to detect and map invasive blooms for their following collection by personnel in charge. For example, researchers in [9] published their progress on the development of a trimaran-shape USV that follows a route and registers data using a SONAR sensor and a downward-facing camera. While the SONAR data was used to calculate water depth and vegetation density, the camera collected visual information for post-analysis about areas with and without *european water-milfoil*. Frequently, aquatic vehicles are designed to work standalone, however, authors already published about the fusion of simultaneous data coming from an UAV and USV to better detect invasive weed [8], support data collection [13], and enhance path planning [4]. In contrast, authors in [12] posit that both UAVs and USVs have sight constraints and rely more on a complete aerial approach by combining remote sensing and UAV imaging to map floating weeds, such as *ribbonweed*, *pondweed*, *parrot's feaher*, and *cumbungi*.

The election of sensors for aquatic weed detection varies because some floating algae or weeds can be easily distinguishable from long distances, while others can be partially visible or fully submerged. Table 1 shows the preferred sensing methods by researchers according to the target weed species. It is worth noting the "Weed type" column considers the categories stated by McNabb and Anderson [2]: submerged aquatic vegetation (SAV), emergent and floating plants, and algae. When it is about SAV detection, researchers use hydroacoustic imaging [11], multispectral imaging [8], underwater cameras [9], and SONAR sensors. In contrast, detecting algae as well as floating and emergent weeds often entails the use of regular RGB cameras. For example, authors in [20] propose to use an RGB camera mounted in a Raspberry Pi Model-B to find regions with floating weed. Finally, authors in [12] and [21] propose tools to leverage remote sensing data to detect and map the appearances of *ribbonweed* and *sargassum*.

Automatic floating weed collection, as a particular research subject for its connection and potential impacts on aquatic life, still demands more research (as, certainly, many more related topics to inland water quality management). In contrast, extensive research has been carried out on robotized debris collection in recent years. For example, two unmanned watercraft, the *WasteShark* (Drone Solutions) and *Clearbot* (Open Ocean Engineering) devices are commercially available for trash collection in water bodies [22]. Nowadays, maximum extra payload, autonomy, operation duration, and collection approach are among some factors that differentiate one device from the others. For example, *WasteSkark* is equipped with route planning algorithms and optical recognition and is capable of identifying and collecting up to 350 kg of floating garbage. Once it cleans an area, *WasteSkark* travels to its base of operations, where it unloads the collected waste and recharges its batteries for the following 16 hours. Moving on, *Clearbot* has the capacity to collect up to one metric ton of plastic per day by using a conveyor belt. These approaches could serve as a good basis for designing better autonomous navigation and floating weed collection systems in the future.

C. AUTONOMOUS DRIVING APPROACHES FOR UNMANNED SURFACE VEHICLES

Approaches for USV's navigation systems development can be organized in two groups: self-relying vehicles and vehicles accompanied by a different sensing modality for better path planning, e.g. aerial imaging and remote sensing. Since the latter usually limits the autonomy of a USV, the present research focuses on building a self-sufficient vehicle that leverages its sensors to determine the best route to follow. Therefore, we orient the literature review towards sensors that detect the current location and methods to estimate the correct steering wheel angle.

Nowadays, we can distinguish three main approaches concerning steering angle prediction: image processing-based, end-to-end learning-based, and deep reinforcement learning-based. Even though the first one has meaningful results in controlled scenarios, it has high failure risks in real scenarios due to the uncertain nature of water bodies. In contrast, end-to-end learning can generalize for diverse scenarios and provide a good estimation of the heading angle with a reasonable computational cost [23]. The main drawback of this method is the extensive datasets it requires to achieve good results [24]. Still, there are several available simulation environments, and currently, it is possible to develop one that is specifically suited to a project's needs. Therefore, we selected this approach for the USV proposed in the present research. It is worth mentioning that recent investigations on the deep reinforcement learning-based approach for autonomous vehicles have obtained compelling results, and it is a research thread worth exploring for future USVs [25].

Compared with urban environments, self-driving vehicles are considered to be more reliable to operate on water bodies' surface due to the fewer obstacles. Nevertheless, navigating autonomously entails several challenges such as aquatic fauna and vegetation, shorelines, manned ships, and most importantly, instability and imprecise position estimation. Most USV proposals depend on depth measurement equipment to detect potential collisions in long distances (e.g. AIS, Radar, GNSS) and short distances (e.g. RGB or spectral cameras, LiDAR, stereo vision) [22]. Other unmanned ships make use of SONARs [9] or light sources to measure depth in water and detect the presence of vegetation or other obstacles. While the main function of sonar sensors is to emit sound waves and measure the time taken for the echoes to return, light sensors are used to estimate water depth and potential obstacles based on the amount of light that passes through a water body.

III. DESIGN AND IMPLEMENTATION OF THE UNMANNED SURFACE VEHICLE

This section describes the materials and methods used when prototyping the USV. Both hardware and software components were implemented, such that the USV could successfully collect duckweed in inland water bodies. The prototype was designed to work under ideal duckweed overgrowth conditions, such as still water, excessive minerals and nutrients, and low wind. These conditions, as described in [26], ensure minimal environmental disturbances that could significantly impact the USV's maneuverability.

A. MECHANICAL DESIGN

First of all, we considered the operation conditions to design the mechanical components of the proposed prototype. For example, the vehicle must have been capable of collecting duckweed in calm water bodies. We used the 3D CAD software SolidWorks to design the vehicle's prototype ensuring the feasibility of the manufacturing and proper assembly of the components. The USV, illustrated in Figure 2, was designed with a catamaran shape and incorporates two floaters on the sides to ensure navigability, as indicated in (1). Additionally, we embedded two compartments within the floaters to house the prototype's batteries (7). These are equipped with protective covers to ensure water resistance. Finally, 2 DC motors were embedded within fins placed at the bottom of each floater, as shown in (8). This configuration eliminates the need for a rudder and increases maneuverability to the joint operation mode of both propellers.

Both floaters are bonded with a main body (6), which has a net in the back to permit the USV collect and transport duckweed when it passes through the USV's interior (5).

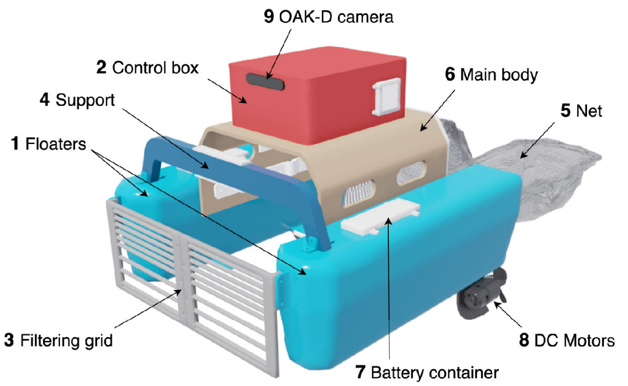


FIGURE 2. **A visual illustration of the solidworks (CAD) design of the USV.**

We included extra protection to both DC motors to avoid potential collisions with the net and added a filtering grid in (3) to prevent access to other types of aquatic wastes (e.g. debris, trash, plastic, or other floating plants). Moreover, we included a box on the top of the main body to safeguard the electronic components (2). This container has apertures on the front and on both sides to install the OAK-D camera and two cooling fans, respectively. Both side vents were designed considering potential splashes or water infiltration. Moreover, we also added support elements to reinforce the joint between both floaters and the main body (4).

B. ELECTRONIC DESIGN

The electronic design considered two sets of components. The first group was focused on the electronic and motion components of the USV, while the second group was geared towards a portable remote control. This was used to operate the vehicle when manual control was necessary. Consequently, both the USV and the remote control were developed as separate subsystems. A detailed description of the components within each subsystem is presented in Table 2.

Figure 3.A illustrates the interconnection of the components installed inside the remote control. In this case, an Arduino Mega 2560 was employed as the main processor for data flow between the vehicle and a ground station. A joystick was used for manual control of the vehicle, along with a set of switches and indicators to determine the operating mode (e.g. manual, autonomous). Additionally, a GPS module was responsible for registering the position of the ground station and generating a comparative signal with the vehicle's location. Moreover, an Xbee 3 RF module was utilized to transmit the movement commands to the vehicle through the Zigbee protocol with a range of 1,200 meters under ideal conditions. A Bluetooth module was also integrated to transmit information from the remote control. Data received through both the Xbee 3 RF and Bluetooth modules was stored for post-processing and analysis. Finally, a set of lithium batteries, along with a voltage regulator, served as the power source.

On the other hand, Figure 3.B displays the components of the USV. In this configuration, a Jetson Nano served as the main processing unit and ensured all navigation data was received from the remote control and transmitted appropriately to the rest of the components. We included an OAK-D camera to acquire RGB and depth images, as well as bounding boxes enclosing duckweed colonies' appearances. Furthermore, we installed a GPS module to help us delimit the vehicle's automatic operation. The USV's control system was installed to relay movement commands to a Raspberry Pi Pico connected via USB, which in turn transmits them to both thrusters through two high-power H-bridges. Finally, two LiPo batteries provided power to all components inside the USV using voltage regulators to meet the individual needs of each component.

C. MANUFACTURING AND ASSEMBLY

The design detailed above and a preliminary material comparison encouraged us to choose 3D printing for manufacturing most of the prototype's components. Specifically, we selected PLA (Polylactic Acid) due to its ease of printing, lightweight feature, and feasible conversion from CAD format to G-code using PrusaSlicer. In addition, we confirmed PLA was also suitable to resist water pressure and let the USV navigate. Although the modular nature of 3D printing made the ensemble process straightforward, the design required modifications and improvements that meant re-printing some components. Yet, this minimized material waste while optimizing the device's functionality. Due to the dimensions of the floaters, they had to be printed in two parts; therefore, water-resistant silicone was used to join both pieces. Furthermore, plastic spray paint was applied to the entire structure, thereby sealing any porosities that may have resulted from the 3D printing process and ensuring complete impermeability and proper flotation of the vehicle.

For the prototype assembly, zinc-coated bolts and nuts were used to enhance water resistance. Such mechanical joints allowed for easy assembly and disassembly of the robot's parts, which was convenient when integrating any additional component. The supports added in the last prototyping iteration provided a more robust structure for holding the superior container and the floaters, thereby increasing the overall strength of the vehicle. All electronic components were interconnected using 16AWG gauge wires, ensuring that they could support the current flow from the batteries. We used heat-shrink tubing and silicone coatings to protect wires in contact with water and ensure electronic components were well-positioned.

D. MOTION LOGIC SYSTEM

We implemented Robot Operating System (ROS) for the USV's motion and logic system due to its versatility to integrate and communicate with sensors and actuators.

TABLE 2. List of hardware components used in the unmanned surface vehicle and remote control.

	List of Components				
	Part	Purpose	Characteristics	Size (mm)	Energy Consumption
USV	Jetson Nano	Procesing	Quad Core ARM A57, 4GB 64 bit LPDDR4, 128-core Maxwell	100 x 80 x 29	5V, 2A
	Raspberry Pi Pico	Control	Dual-Core ARM Cortex M0+ RP2040, 264KB SRAM, 2MB Flash Memory	51 x 21 x 1	5.5V, 2.5A
	OAK-D Camera	Detection	RGB and Stereo Camera, IMX378 12 MP	110 x 54.5 x 33	5V, 0.5A
	GPS GY-NEO6MV2	Localization	UART, -161 dBm sensitivity	25 x 25 x 1	3.3V to 5VDC, 0.05A
	Xbee 3 RF	Communication	Up to 1,200m of range, 8dBm Transmit	13 x 19 x 1	3.3 V, 0.04A
	LiPo Batteries	Power	11.1V voltage, 2,200mAh Capacity	86.11 x 33.02 x 17.78	N/A
	Voltage Regulator	Stabilization	DC-DC Boost 150W, 10-32 V to 12-35V	65 x 56.5 x 23	N/A
	H-Bridge Driver	Control	6V to 27V voltage, max current 43A, PWM or level control, 2 channels	43 x 43 x 27	N/A
	DC Motors	Movement	14,600 RPMs, 1.5 kg.m torque	45 x 67	24V, 4.7A
Remote Control	Arduino Mega 2560	Control	ATmega2560, 8KB SRAM, 256KB Flash Memory	101.52 x 53.3 x 1	5V, 0.05A
	Joystick Module	Movement	Biaxial Joystick XY, 2 x 10Kohm potentiometers, NO button	40 x 26 x 32	5V
	Xbee 3 RF	Communication	Up to 1,200m of range, 8dBm Transmit	13 x 19 x 1	3.3 V, 0.04A
	GPS GY-NEO6MV2	Localization	UART, -161 dBm sensitivity	25 x 25 x 1	3.3V to 5VDC, 0.05A
	Bluetooth Module HC-05	Communication	Up to 10m of range, 1.3Mbps	37 x 16 x 1	6V, 0.05A
	Lithium Batteries	Power	3.7V voltage, 4,800mAh	18 x 65	N/A
	Voltage Regulator	Stabilization	Step Down regulator, 2 to 37V, 3A	43 x 20 x 14	N/A

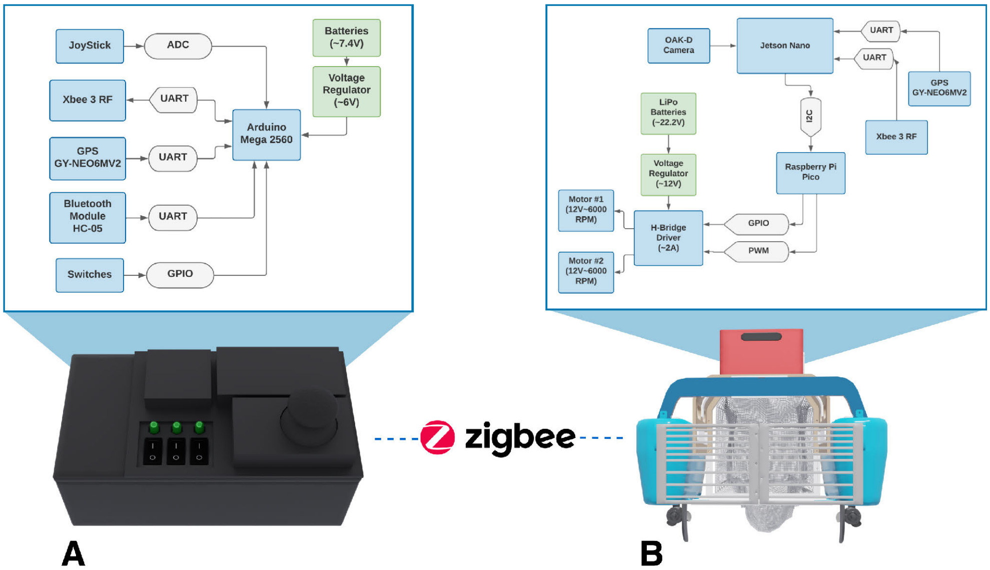


FIGURE 3. Block diagram of the components used in the USV and remote control.

Although ROS was used mainly for data reception from the Xbee 3 RF module, GPS module, and the OAK-D camera, it also incorporated the necessary algorithms for the SAE system, detailed in Section IV. Moreover, ROS also communicated inference results with the actuators, which in turn required the implementation of Micro-ROS to send the motion logic to both motors. The diagram of topics and nodes used in the ROS implementation is shown in Figure 4, from where each subsystem is detailed in the following sub sections.

1) MICRO-ROS IMPLEMENTATION

For the implementation of the motion logic, it was first necessary to establish communication between the Raspberry Pi Pico and the Jetson Nano Card. This connection was achieved by configuring the information transmission between both devices via UART and utilizing `micro-xrce-dds-agent` to manage this data. Within this connection, a topic is generated for the reception of the desired velocity information from the Jetson Nano, and this information is sent to the Raspberry Pi Pico in the form of

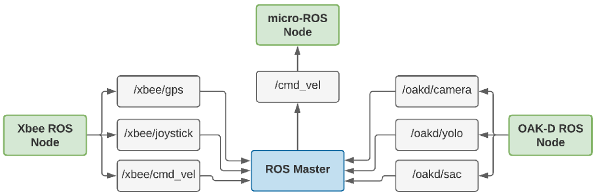


FIGURE 4. Block diagram of the ROS architecture.

TABLE 3. Modalities of operation and motor PWM configurations.

Modality (Velocity)	Condition (Direction)	Motor #1 PWM	Motor #2 PWM
Linear Velocity in X	> 0	Forward	Forward
	< 0	Backward	Backward
Angular Velocity in Z	> 0	Forward	Backward
	< 0	Backward	Forward

linear and angular velocity (along the X, Y, and Z axes). Then, these values are packaged in the topic `/cmd_vel`.

The code for reading the angular and linear velocity information was implemented using C++ since the Raspberry Pi Pico only supports this functionality in this programming language. The Equation 1 and Equation 2 were used to translate these linear and angular velocities into PWM values for each of the motors individually.

$$PWM_{left} = K_{PWM} \times (v - L \times \frac{\omega}{2}) \tag{1}$$

$$PWM_{right} = K_{PWM} \times (v + L \times \frac{\omega}{2}) \tag{2}$$

Equations 1 and 2 describe the PWM value as a function of the linear velocity v (movement forward or backward) and angular velocity *omega* (rate of turn around its own axis). Additionally, two terms are added where the value of L is given by the distance between the motors. The value of K_{PWM} is determined by a constant that scales the PWM value to the desired value. In this case, this value was determined experimentally so that a linear velocity of $1\frac{m}{s}$ equates to the maximum PWM value on both motors. These equations were derived from those determining the turning speed of each motor in a differential robot, which is similar to the configuration of the proposed vehicle.

To determine these turning modalities, we assumed that pin A is on and pin B is off when turning forward. Likewise, when turning backward, pin A is off, and pin B is on. This configuration considers that there are two PWM values for each motor and is used to demonstrate how the direction of rotation of both motors varies to determine the vehicle's state of motion based on the previously determined linear and angular velocities, as shown in Table 3. Therefore, all possible operating states were considered.

2) Xbee ROS NODE

As previously outlined, the logic for the movement of each motor was embedded within the Raspberry Pi Pico, which subscribes to the `cmd_vel` topic. The data sent to this was processed in two ways: firstly, in manual mode, where they were extracted from the data received via the Xbee 3 RF module, and secondly, according to the states of motion defined by the SAE system (described in Section IV). For manual operation, data was transmitted from an Arduino Mega 2560, where the X and Y position data were encoded into a string to be decoded in the USV. These data values were then decoded, and depending on the X and Y position of the Joystick, a linear or angular velocity for the robot was determined and limited to $1\frac{m}{s}$ for linear velocity and $1\frac{rad}{s}$ for angular velocity. These commands were subsequently published to `/cmd_vel` and received by the Raspberry Pi Pico.

Furthermore, it is important to note that the Joystick commands were also published on the `/xbee/joystick` topic, so the Joystick could be used for other purposes than only manual vehicle control. Additionally, owing to the integrated sensors within the vehicle, the topic also gathered data from the GPS module. This module sent encoded NMEA-type signals through its USB port via UART. Also, these signals were decoded to produce longitude and latitude data, which were then published on the `/xbee/gps` topic. This had the aim to use the information for vehicle tracking and for delimiting the USV's working area.

3) OAK-D ROS NODE

We implemented an OAK-D camera as the primary sensor for the USV's context data acquisition. Not only does the camera capture RGB and depth images, but it also has a System on Module (SoM) of 4 Tera Operation Per Second (TOPS) that permits the embedding of an object detection model. Such a feature let us deploy a duckweed detection model inside the camera and save computational power. The obtained duckweed detections were generated at an approximate speed of 2 frames per second (FPS), which was relatively reasonable given the vehicle's data collection speed. The RGB and depth images were published on the `/oakd/camera` topic, making this information accessible from the main operating script. Similarly, duckweed detections were published on the `/oakd/yolo` topic, where information about bounding boxes, the number of detections, and their confidence level was stored. Furthermore, steering angles obtained with SAC (further detailed in Section IV-C) were published on a new topic named `/oakd/sac`, making this result available for the main ROS script.

IV. IMPLEMENTATION OF THE STEERING ANGLE ESTIMATION SYSTEM

A. SYSTEM DESIGN

The principal purpose of the SAE system is to provide navigation autonomy and collision avoidance capabilities to the USV. Four self-driving approaches, named $M1$, $M2$, $M3$, and $M4$, were developed gradually to obtain the final SAE's architecture. The approaches' composition is summarized in Table 4, and Figure 5 depicts the overall architecture of the

TABLE 4. Self-driving modalities explored in the investigation.

Modality	Duckweed detection	Momentum	SV-based SAC	DL-based SAC
$M1$	yes	no	no	no
$M2$	yes	yes	no	no
$M3$	yes	yes	no	yes
$M4$	yes	yes	yes	no

SAE system. In this figure, the section on the left represents an OAK-D stereo vision camera, which obtains RGB and depth information from the USV's environment. On the other hand, the section in the middle shows the SAE system, which is deployed inside a Jetson Development Kit to process the obtained RGB images, bounding boxes, and depth maps to calculate the angle α for the next USV movement.

Initially, training a duckweed detection model let us propose the $M1$ self-driving mode, which relied uniquely on duckweed detection to infer the next best angle α. This was achieved by averaging the horizontal coordinates of the detected bounding boxes using Equation 3. There, d represents the number of horizontal partitions, w is the input RGB image's width, and n is the number of detected bounding boxes. Additionally, x_i and z_i represent the x coordinates of the top left and bottom right of each bounding box, respectively. Given that $M1$ caused unpredictable, inconsistent navigation, we noted the importance of evolving the model, which led us averaging the current angle α with the immediate nine previous angles. We named this calculation *momentum*, and its formula is given in Equation 4, where B and m were defined experimentally. Moreover, we denominated the combination of $M1$ and *momentum* as $M2$.

$$\alpha = \left| \frac{d}{w} \times \frac{1}{n} \times \sum_{i=1}^{n} \left(\frac{z_i - x_i}{2} \right) \right| \tag{3}$$

$$\alpha(\mathrm{n}) = \left| \alpha \cdot (\mathrm{n})^{\mathrm{B}^1} + \ldots + \alpha \cdot (\mathrm{n} - \mathrm{m})^{\mathrm{B}^\mathrm{m}} \right| \tag{4}$$

$$f(t, c) = \begin{cases} \alpha = 0 & \textit{if } t = 0,\ c = 3 \\ w_1 = 1, w_2 = 0 & \textit{if } t = 0,\ c \neq 3 \\ w_1 = 0, w_2 = 1 & \textit{if } t \neq 0,\ c = 3 \\ w_1, w_2 & \textit{otherwise} \end{cases} \tag{5}$$

Later, to provide collision avoidance capabilities to the USV, we designed the $M3$ self-driving mode by appending a Deep Neural Network model (DNN) to $M2$ to classify the next best steering angle according to the USV's visual perception. This approach, named Steering Angle Classification (SAC), had the purpose of detecting and avoiding environmental disturbances, such as boats, ducks, and buoys, among other aquatic objects. We contrasted $M3$ with a fourth self-driving mode, named $M4$, which applies SAC by replacing the DNN with stereo vision (SV). Then, we averaged the α_1 and α_2 angles coming from SAC and Equation 3 weightily using the variables w_1 and w_2. While the values of w_1 and w_2 were determined experimentally, we used Equation 5 to pre-establish their values according to the detections sent by the inner modules. In this equation, t represents the total number of duckweed bounding boxes, and s represents the steering angle detected by the SAC algorithm. For both $M3$ and $M4$, the result is passed to the *momentum* module, shown in blue in Figure 5, to obtain the optimal α.

For all self-driving approaches, the best next angle α is communicated to the H Bridge Driver, which in turn transforms the values and activates the respective motors to rotate the USV. The SAE system repeats this process while the vehicle is in operation, and GPS coordinates are used to limit the navigation area. While the physical inspection area's borders were detected reading the GPS coordinates, we managed to mimic the same behaviour in the simulator using red bands; see Figure 6. When the USV reaches a border, it is programmed to turn around by selecting a random angle in the range [0°, 180°] contrary to the border. Finally, the vehicle is programmed to send notifications when it is about to run out of power or if the net containing duckweed is full.

B. DUCKWEED DETECTION

We implemented versions 5 and 8 of the Yolo architecture to detect duckweed colonies over water bodies [27], [28]. Given a series of challenges for the detection task, we performed extensive experiments augmenting and enhancing the initial dataset. Mainly, we balanced the frequency of images with close and far duckweed colonies. Although duckweed is easily distinguishable in short distances, it looks like a grass field when seen from a long distance. Both dataset preparation and model implementation are explained in the next subsections.

1) DATASET PREPARATION

Research on floating weed detection is narrow, and related image datasets are almost non-existent. In the case of duckweed, we struggled to find datasets from a surface perspective, which made us collect our dataset. Because of the challenges of collecting diverse images in situ, we opted to scrape images from the Internet using Selenium and Python. The first part of the dataset comes from Google Images and was found using the terms "lemna minnor", "duckweed", "lenteja de agua", "lago titicaca", "laguna alalay", among other terms. It is worth mentioning that we made sure to download images with Creative Commons licenses that permit reutilization for non-commercial purposes. We achieved this by selecting the option "Creative Common Licenses" on the image searcher's interface with Selenium. The second part of the dataset comes from the Global Biodiversity Information Facility Repository (GBIF) [7], which is an international network and data infrastructure funded to provide open access to data about all types of life on Earth. Similarly, we used Selenium and Python to collect

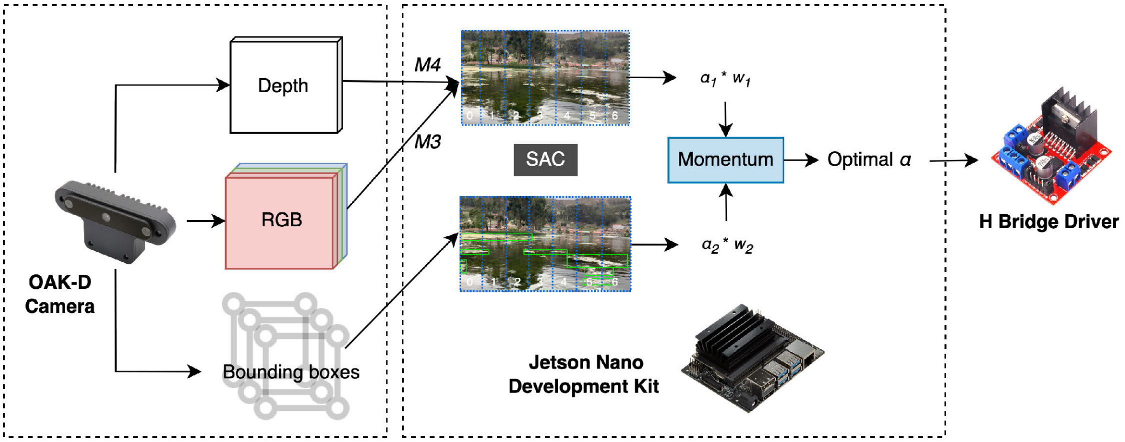


FIGURE 5. The overall architecture of the SAE system.

TABLE 5. Quantity of images in each POV class for the base dataset and augmented versions.

Dataset	Close	Near Close	Near Wide	Wide	Empty
Base	439	1,291	993	699	578
Virtual	614	0	0	318	448
Augmented	947	709	1,007	983	974
Total	2,000	2,000	2,000	2,000	2,000

images with the terms “lemna gibba”, “lemna minor”, “lemna trisulca”, “spirodela polyrhiza”, among others.

Once we obtained 4,341 suitable images, we started normalizing them with respect to their format, color space, and dimensions. Moreover, initial manual reviews and Exploratory Data Analysis (EDA) let us discard samples with blur, oversize, and low illumination. We made sure that all images contained at least one duckweed colony seen from a close or distant point of view (POV), which resulted in a dataset of 3,335 images. Then, we used the Roboflow platform to label the images and included 665 pictures of water bodies without duckweed. Furthermore, because we identified miss-recognition between samples captured from long and short distances, we labeled the dataset according to four categories: close-view, long-view, near-close view, and near-long view. In order to increase the diversity of the training data, we made use of the simulation environment (detailed in Section IV-C) to collect synthetic samples. Additionally, we applied random augmentations (rotations, horizontal and vertical flipping, illumination variation, and noise incorporation) to the images to balance the POV frequencies. Table 5 shows the POV frequencies in the resulting final dataset, which contains 10,000 instances.

2) MODEL IMPLEMENTATION

Deep neural networks have demonstrated better generalization benefits for conventional tasks such as object detection, image segmentation, and classification. However, the more complex the annotations required for the task are, the more human effort and time it takes to complete a dataset to train a model. From this point of view, we defined the duckweed detection task to be a specific application of object detection. Still, we remark on the importance of exploring the segmentation task in future research work. We implemented Yolov5 and Yolov8 because of their performance with regard to the COCO benchmark dataset [28]. Both versions of Yolo are available in five versions for retraining with a custom dataset: n, s, m, l, and x. The codes denote the number of parameters and model footprint, with n as the lightest and x as the heaviest. The model election was governed by the lightest footprint as opposed to the highest precision. As such, the Yolov5s and Yolov8s versions were used in this project.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP+FP+TN+FN}} \tag{6}$$

$$\text{Sensitivity(Recall)} = \frac{\text{TP}}{\text{TP+FN}} \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{8}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \tag{9}$$

$$\text{F1} - \text{score} = \frac{2 \times TP}{2 \times TP\text{+FP} + \text{FN}} \tag{10}$$

Our evaluation algorithm processed the detected bounding boxes to convert them into 2D binary masks, where pixels inside bounding boxes were assigned 1. We managed to obtain a confusion matrix by quantifying the correct matches, which in turn let us count the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Then, these values allowed us to calculate the Accuracy, Sensitivity (Recall), and Specificity values according to Equations 6, 7, and 9, respectively. Moreover, we calculated the Average Precision (AP) per class by finding the area under the precision-specificity curve:

$$AP = \int_0^1 p(r)\,\mathrm{d}r. \tag{11}$$

where p is the proportion of true positives (precision), and r is the proportion of true positives out of the possible positives (recall). Both variables are always between 0 and 1. Therefore, the average precision per class falls between 0 and 1 as well, and it helps calculate the mean Average Precision metric (mAP).

Besides these metrics, we put further consideration on the Intersection over Union (IoU) metric because of the higher importance of detecting duckweed appearances, as opposed to assessing if pixels were assigned to the correct bounding boxes. IoU quantifies the similarity between a predicted mask and a ground truth mask. For the project, IoU was defined as in Equation 12, where G and P were obtained by converting labeled and predicted bounding boxes of the same image into binary masks, respectively.

$$IoU = \frac{|G \cap P|}{|G \cup P|} \tag{12}$$

C. STEERING ANGLE CLASSIFICATION

As previously stated in Section II-C, developing an autonomous control system for unmanned vehicles is always a challenge because of the many exceptions that happen in real environments. While some approaches make use of multiple sensors for collision avoidance [9], others attempt to map the environment first for better path planning [4], [11]. In the present project, we developed a prototype of USV to be deployed in any environment without prior exploration. Therefore, we researched the approaches $M3$ and $M4$ for the USV to avoid collisions using end-to-end learning and stereo vision. These approaches will be explained and compared in the following sections.

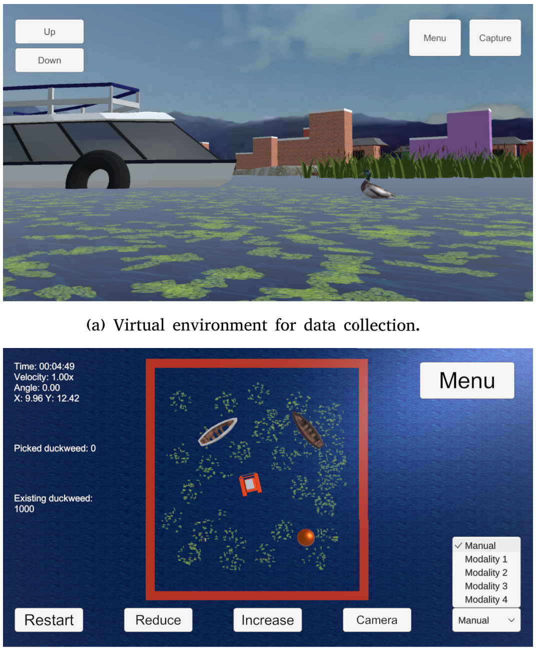
1) END-TO-END LEARNING

For the $M3$ approach, we explored end-to-end-based models that could let the USV avoid collisions by mapping the correlations between RGB images and steering angles. End-to-end learning has proved to be efficient in generalizing diverse scenarios and obstacles with a reasonable computational cost. Nevertheless, its main drawback is that it requires large annotated datasets. Given that data collection in situ was a limiting factor for the project, we created a simulator inspired by the Udacity Self Driving Car Nanodegree emulator using Unity, Blender, C# and Python. As shown in Figure 6, this simulator resembled Lake Titicaca and helped us perform two core tasks: collect images and perform testing with the self-driving modalities. Specifically, the simulator let us collect 1,380 images with duckweed colonies and 37,846 images of manually controlled trips.

As soon as we finished recording trips in visual and numeric format, we preprocessed the dataset by flipping the images and angles. Then, we started training generic regression models based on the ResNet34 architecture. Yet, the initial regression metrics let us note quickly the importance of balancing underrepresented angle ranges and transforming the task into a classification problem. Consequently, we split the angle range into seven and obtained the respective bins to categorize each sample's angle, which in turn resulted in the steering category distribution shown in Table 5, in column "Base". This decision was also supported by the fact that the movement of a USV changes slowly compared with unmanned aerial or terrestrial vehicles. Finally, we performed augmentations to obtain 7,000 samples per steering category, adding up to a total of 49,000 instances.

TABLE 6. SAC dataset evolution with respect to steering angle categories.

Class	Angle	Base	Flipped	Augmented	Total
0	-0.45	747	1,517	5,483	7,000
1	-0.30	798	1,713	5,287	7,000
2	-0.15	1,850	3,876	3,124	7,000
3	0	30,741	7,001	0	7,001
4	0.15	2,026	3,876	3,124	7,000
5	0.30	915	1,713	5,287	7,000
6	0.45	770	1,517	5,483	7,000



Time: 00:04:49
Velocity: 1.00x
Angle: 0.00
X: 9.96 Y: 12.42

Picked duckweed: 0

Existing duckweed:
1000

(a) Virtual environment for data collection.

(b) The robot inside the environment for self-driving testing.

FIGURE 6. Our virtual environment for synthetic data collection and autonomous driving testing.

Despite the application of data augmentation, our initial classifiers failed to recognize angles reliably. Navarro et al. [23] highlighted the need for large datasets to train steering angle detectors based on end-to-end learning. Therefore, we opted to apply transfer learning by modifying the last layer of five generic models available in the PyTorch's Torchvision package [29]. Specifically, we used the implementation of ResNet-50, ResNet-101 [30], EfficientNet [31], ConvNeXt [32], and ResNeXt-101 [33]. These are provided with their corresponding pre-trained weights obtained using the ImageNet dataset. Then, we replaced their last layer with a layer of 7 output nodes and applied a softmax activation layer. Furthermore, to fine-tune all models, we split the dataset into 34,301 for training (70%), 9,800 for validation (20%), and 4,900 for testing (10%). For the models to converge efficiently, we started testing hyperparameters with only one model and used similar hyperparameters once we obtained a suitable combination.

To measure the performance obtained by the models, we used a confusion matrix per validation or testing stage. The predicted and ground-truth labels let us quantify the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Then, these values let us calculate the Accuracy, Sensitivity (Recall), Specificity, and F1-score according to Equations 6, 7, 9, and 10, respectively.

2) STEREO VISION

Moving on, the stereo vision-based approach, namely $M4$, split the depth maps coming from the OAK-D camera in seven vertical regions. Then, each region was used to calculate the average depth and select an optimal angle α, such that regions with potential obstacles could be avoided. A graphical representation of the approach is illustrated in Figure 7, where the effective depth detection range is shown in red. Similarly to $M1$, SAC based on stereo vision outputs a discrete number in the range [0, 6] as the optimal angle. Then, this is passed to *momentum* to obtain the final optimal angle.

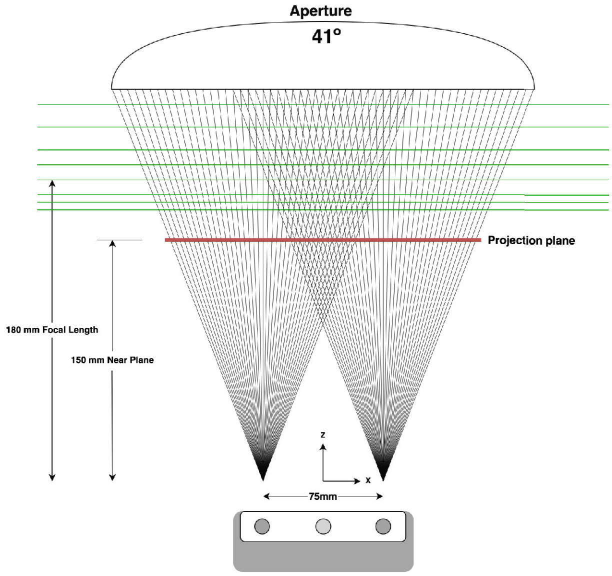
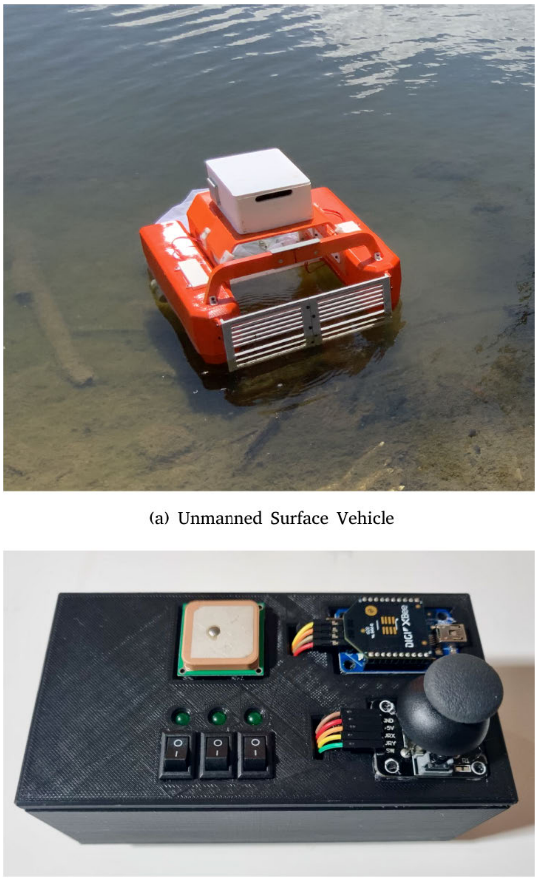


FIGURE 7. Integral disparities in 3D space for the OAK-D camera.

V. EXPERIMENTAL RESULTS

A. THE UNMANNED SURFACE VEHICLE

To test the buoyancy, resistance, and navigability of the USV prototype, we conducted a series of field experiments at two different sites. Initial tests were performed on April 6th, 2023 at the Olympic Swimming Pool in Alto Obrajes, La Paz, Bolivia. There, we aimed to test the USV's behavior in calm water conditions when controlled via the Zigbee protocol using the remote control. Problems with the collection net and buoyancy were identified and managed properly for the next testing phases. A demonstration video of these tests is available in [34]. The second testing phase was carried out on October 14th, 2023 in Lagoon Achocalla, La Paz, Bolivia, where the environment resembles Lake Titicaca. The goal of these tests was to evaluate not only the USV's buoyancy on wavy water but also navigability with additional factors such as local fauna and flora, among other elements. A demonstration video of these tests is available in [35]. Additionally, a pair of pictures illustrating the final USV and Remote Control prototypes is shown in Figure 8.


(a) Unmanned Surface Vehicle

(b) Remote control

FIGURE 8. Prototypes of the USV and remote control.

The present research has successfully developed the necessary hardware for automated floating invasive plant removal. We consider the testing and enhancement of the USV as an ongoing project given the challenges to test in situ. For instance, the southern hemisphere, and specially Lake Titicaca, are currently at devastating drought due to frequent heat waves that have caused record low water levels [36]. On the other hand, we carried out exhaustive experiments to evaluate the performance of the SAE system inside the simulator. These experiments and SAE submodules' performance will be discussed in the next sections.

TABLE 7. Comparison of results obtained with Yolov5s and different versions of the base dataset reported in Table 5.

Categories considered	Precision	Recall	mAP	IoU
duckweed	0.481	0.375	0.386	0.749
duckweed (augmented)	0.571	0.419	0.408	0.788
close + empty	0.910	0.840	0.910	0.934
wide + empty	0.660	0.520	0.586	0.917
close + wide + empty	0.708	0.644	0.698	0.674
all (augmented)	0.452	0.412	0.396	**0.821**

B. DUCKWEED DETECTION

The overall architecture of the SAE system relies on the precision of the duckweed detector. As mentioned above, although generic object detectors feature great generalization capabilities, we initially identified problems when fine-tuning detectors using image samples captured from distant and close POV. This was because a duckweed colony resembles a field of grass when seen from a distant POV. For that reason, our initial experiments were oriented to test the scale-invariant properties of Yolov5s. We downloaded weights from the Ultralytics repository [27] and retrained the model with new dataset versions obtained from the Base Duckweed Dataset described in Table 5. The resulting IoUs let us note that augmenting samples considering the POV categories (shown in the 6^{th} row) was a better approach than augmenting the dataset in general (shown in the 2^{nd} row). Additionally, it is worth mentioning that the detectors obtained compelling metrics when we focused on only one POV category at a time (shown in the 3^{rd} and 4^{th} rows). To obtain the results in Table 7, we used 10 epochs, the ADAM optimizer, 0.01 learning rate, and 16-samples batch-size as hyper-parameters to evaluate all models fairly.

Moving on, we comparatively tested the combination of the “Base” dataset, named “Real” dataset, and the dataset obtained with the simulator, named “Virtual” dataset. We also augmented the combined version to equalize the number of instances per POV category. Table 8 lists IoU, mAP, and Accuracy values of custom-trained Yolov5s and Yolov8s tested with a subset composed of only real samples. It is interesting to see how Yolov8s trained with the “Augmented” dataset (last row) outperformed other models with a 0.894 IoU. In the same table, the last two columns show the Average Duration (AD) time and Effective Cleaning Percentage (ECP) when testing the models for the $M1$ self-driving mode 10 times. While AD was calculated by measuring the total time it took the USV to clean duckweed from an area of 24.8 m^2, ECP measured the difference between the initial duckweed colonies and the remaining ones after 20 minutes. An AD of 1,093 for Yolov8s shows the relevance of detection precision in reducing the weed cleaning time.

C. STEERING ANGLE ESTIMATION

The progressive development of the SAE system permitted explore 4 driving modalities, as defined in Table 4. While $M1$ and $M2$ aim to move forward by tracking duckweed, $M4$ and $M3$ also avoid potential collisions using stereo vision and end-to-end learning, respectively. In contrast to the stereo vision approach, applying deep learning (DL) to steering angle classification required the collection of a dataset as well as the implementation of classification models. Table 9 lists the Accuracy, Recall, and F1-score metrics of five classifiers that were fine-tunned with the “Augmented” dataset detailed in Table 6. EfficientNet outperformed the rest of the models with 0.831, 0.972, and 0.831 values for accuracy, recall, and F1-score, respectively. Although ResNet50 obtained lower precision results, it got better efficiency in terms of FPS and necessary megabytes to store the model.

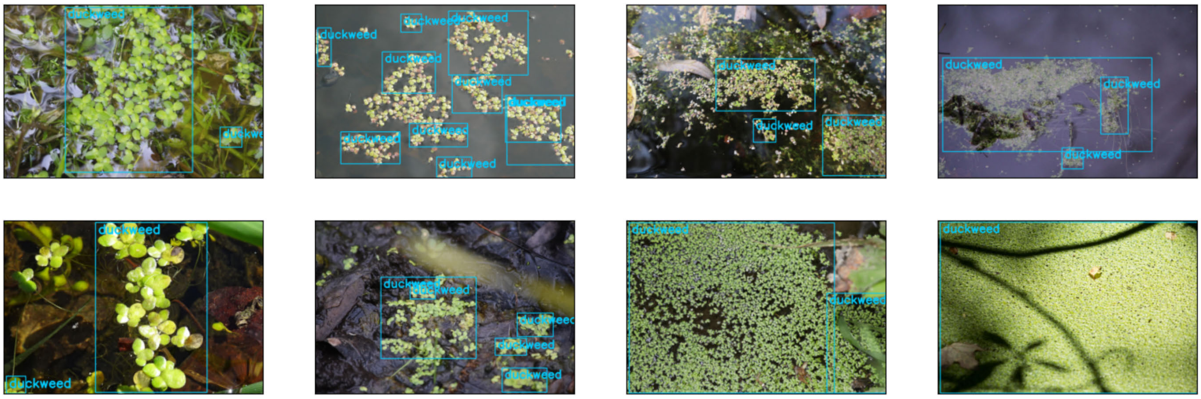
All, $M2$, $M3$, and $M4$ make use of the *momentum* module to define the best angle according to Equation 4. Experimental testing of 10 repetitions let us find $B = 0.3$ and $m = 10$ as the best parameters for Equation 4. This testing phase was performed with the $M2$ modality inside the simulator and let us obtain the average least time (1,064 seconds) and highest cleaning success (98.6%). To evaluate the performance of $M3$ and $M4$ modes, we created a specific section with barriers, duckweed colonies, and obstacles inside the virtual environment; see Figure 6. Then, we varied the weights w_1 and w_2 in the range $[0.1, 0.9]$ and performed 10 executions per combination and self-driving mode. The averaged results are shown in Table 10, from which we can conclude collision avoidance based on stereo vision ($M4$) obtained better performance than the end-to-end learning approach. Since the vehicle completed most of its cleaning tasks efficiently with $w_1 = 0.8$ and $w_2 = 0.2$, we used the same parameters when deploying the SAE system inside the USV.

TABLE 8. Comparison between results obtained with Yolov5s and Yolov8s with the datasets reported in Table 5. We include $M1$ results as for average duration (AD) and effective cleaning percentage (ECP).

	Dataset	IoU	mAP	Acc.	AD (seconds)	ECP (%)
Yolov5s	Only real	0.743	0.643	0.728	1,276.60	71.6%
	Only Virtual	0.819	0.649	0.809	1,251.36	94.7%
	Real + Virtual	0.850	0.878	0.833	1,201.05	89.3%
	Real + Virtual + Data Augmentation	0.863	**0.866**	0.847	1,108.13	**93.2%**
Yolov8s	Only real	0.876	0.778	0.859	1,200.00	91.0%
	Only Virtual	0.874	0.602	0.889	1,185.18	84.3%
	Real + Virtual	0.809	0.690	0.788	1,188.67	91.9%
	Real + Virtual + Data Augmentation	**0.894**	0.799	**0.880**	**1,093.00**	91.6%

TABLE 9. Steering angle classification results.

Model	Acc.	Recall	F1-score	Gflops	FPS	Weight
Resnet50	0.794	0.966	0.794	**4.09**	**765**	**90**
Resnet101	0.812	0.969	0.812	7.8	693	162.7
EfficientNet	**0.831**	**0.972**	**0.831**	24.58	490	203.1
ConvNext	0.736	0.956	0.736	15.36	462	334.2
ResNext	0.824	0.971	0.824	16.41	603	331.9



(a) Close (b) Near close (c) Near wide (d) Wide

FIGURE 9. Detected duckweed colonies using the best performing Yolov8s model. Columns denote the point of view from where the samples were taken.

TABLE 10. Comparison between results obtained with Modality *M*3 and *M*4. AD and ECP stand for average duration and effective cleaning percentage, respectively.

		$M3$		$M4$	
$\mathbf{w_1}$	$\mathbf{w_2}$	**AD (s)**	**ECP (%)**	**AD (s)**	**ECP (%)**
0.1	0.9	1,200	62.24%	1,190.6	75.80%
0.2	0.8	1,200	66.57%	1,174	87.87%
0.3	0.7	1,200	69.63%	1,183.8	84.45%
0.4	0.6	1,200	**73.27%**	1,154.7	84.14%
0.5	0.5	1,200	68.09%	1,152.7	89.33%
0.6	0.4	1,200	66.49%	1,173.8	86.03%
0.7	0.3	1,200	66.45%	1,199.9	85.72%
0.8	0.2	1,200	67.04%	**1,122**	**89.95%**
0.9	0.1	1,200	64.10%	1,194.4	80.20%

VI. CONCLUSION

In this paper, we propose the development of an unmanned surface vehicle that incorporates a steering angle estimation system for the monitoring and removal of floating invasive weeds. Specifically, we study the case of duckweed due to its recent far-reaching ecological and economic implications in Lake Titicaca. The USV proposed in the investigation has been successfully implemented with the required hardware and software capabilities to collect duckweed. Our approach aimed to leverage a bespoke simulator to collect data synthetically and test the removal system exhaustively. Moreover, the simulator let us identify stereo vision-based $M4$ as the most efficient self-driving approach when defining 0.8 and 0.2 for the w_1 and w_2 modules' weights, respectively.

To date, there is no dataset for floating weed detection or end-to-end steering angle classification in inland waters. Therefore, as a result of conducting this research, we also put forward two datasets for the mentioned tasks. The accuracy obtained by EfficientNet (0.831) indicates it was the best-performing model for steering angle classification. Moving on, Yolov8s obtained 0.894 IoU using the dataset that was augmented according to viewpoint. While this strengthens the importance of enhancing the data, it shows that scale, rotation, and light variances can heavily affect the recognition task. Therefore, testing should be performed with different weather conditions, obstacles, and locations for real-world deployment.

The new challenges to control floating weed plagues can be understood by the increasing pollution near inland water bodies around the world. The findings that we have presented here suggest that future versions of the USV should leverage aerial imaging and incorporate path planning and simultaneous localization. This would open doors to develop a multi-USV system to cover longer areas and split the floating weed removal task. Additionally, for further studies, it would be fruitful to incorporate a better collection mechanism on the front side of the vehicle.

ACKNOWLEDGMENT

The proposed approach won the First Global Prize in the OpenCV AI Competition 2022 that was organized by the OpenCV Foundation.

REFERENCES

[1] R. L. Zimdahl, *Fundamentals of Weed Science*, 5th ed. London, U.K.: Elsevier, 2018.

[2] T. McNabb and L. Anderson, "Aquatic weed control," in *Principles of Weed Control in California*. Fresno, CA, USA: Thomson Pubns, 1985, pp. 440–455.

[3] M. Penmetcha, S. Luo, A. Samantaray, J. E. Dietz, B. Yang, and B.-C. Min, "Computer vision-based algae removal planner for multi-robot teams," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 1575–1581.

[4] S. Jung, H. Cho, D. Kim, K. Kim, J.-I. Han, and H. Myung, "Development of algal bloom removal system using unmanned aerial vehicle and surface vehicle," *IEEE Access*, vol. 5, pp. 22166–22176, 2017.

[5] A. Delgado, A. Cotrina, J. Cueto, R. Figueroa, J. Huarcaya, H. Lopez, and L. Andrade-Arenas, "Assessment of water quality in Titicaca lake, Peru using the grey clustering method," *Eng. Lett.*, vol. 30, no. 2, pp. 847–853, 2022.

[6] S. Jabiel. (2022). *Encontraron Miles de Ranas y Peces Muertos en el Lago Titicaca y se Unieron Para Terminar Con la Contaminación*. elDiarioAR. [Online]. Available: https://www.eldiarioar.com/sociedad/medio-ambiente/encontraron-miles-ranas-peces-muertos-lago-titicaca-unieron-terminar-contaminacion_1_9202402.html

[7] GBC. *Global Biodiversity Information Facility*. Accessed: Aug. 28, 2023. [Online]. Available: https://www.gbif.org/

[8] J. E. Perrin, S. R. Jernigan, J. D. Thayer, A. W. Howell, J. K. Leary, and G. D. Buckner, "Sensor fusion with deep learning for autonomous classification and management of aquatic invasive plant species," *Robotics*, vol. 11, no. 4, p. 68, Jun. 2022.

[9] R. Codd-Downey, M. Jenkin, B. B. Dey, J. Zacher, E. Blainey, and P. Andrews, "Monitoring re-growth of invasive plants using an autonomous surface vessel," *Frontiers Robot. AI*, vol. 7, Jan. 2021, Art. no. 583416.

[10] R. Sravanthi and A. Sarma, "Automatic navigation of USV for floating aquatic weed removal in lake," *Adv. Appl. Math. Sci.*, vol. 20, pp. 3205–3218, 2021.

[11] M. Patel, S. Jernigan, R. Richardson, S. Ferguson, and G. Buckner, "Autonomous robotics for identification and management of invasive aquatic plant species," *Appl. Sci.*, vol. 9, no. 12, p. 2410, Jun. 2019.

[12] J. Brinkhoff, J. Hornbuckle, and J. Barton, "Assessment of aquatic weed in irrigation channels using UAV and satellite imagery," *Water*, vol. 10, no. 11, p. 1497, Oct. 2018.

[13] A. H. Göktoğan, S. Sukkarieh, M. Bryson, J. Randle, T. Lupton, and C. Hung, "A rotary-wing unmanned air vehicle for aquatic weed surveillance and management," *J. Intell. Robot. Syst.*, vol. 57, nos. 1–4, pp. 467–484, Jan. 2010.

[14] S. Kumar and S. Sondhia, "Chemical control of duck weed and its effect on water quality and residue," *Indian J. Weed Sci.*, vol. 49, no. 1, p. 105, 2017.

[15] CoastalSpray. *How to Remove Duckweed From a Pond: 5 Tips From the Pros*. Accessed: Oct. 2, 2023. [Online]. Available: https://www.coastalspray.com/5-tips-for-controlling-duckweed/

[16] J. H. Chick and M. A. Pegg, "Invasive carp in the Mississippi river basin," *Science*, vol. 292, no. 5525, pp. 2250–2251, Jun. 2001.

[17] Elastec. *Duckweed Skimmer*. Accessed: Sep. 22, 2023. [Online]. Available: https://www.elastec.com/products/oil-spill-skimmers/weir-oil-skimmers/duckweed-removal-skimmer/

[18] Oilskimmer. *Duckweed Skimmers*. Accessed: Sep. 3, 2023. [Online]. Available: https://www.oilskim.com/oil-skimmers/duckweed-skimmers

[19] Vikoma. *Oil Skimmers for Every Application*. Accessed: Sep. 23, 2023. [Online]. Available: https://www.vikoma.com/Oil-Spill-Solutions/Skimmers

[20] R. Sravanthi and A. S. V. Sarma, "Efficient image-based object detection for floating weed collection with low cost unmanned floating vehicles," *Soft Comput.*, vol. 25, no. 20, pp. 13093–13101, Oct. 2021.

[21] L. Song, Y. Chen, S. Liu, M. Xu, and J. Cui, "SLWE-Net: An improved lightweight U-Net for SARGASSUM extraction from GOCI images," *Mar. Pollut. Bull.*, vol. 194, Sep. 2023, Art. no. 115349.

[22] I. Bae and J. Hong, "Survey on the developments of unmanned marine vehicles: Intelligence and cooperation," *Sensors*, vol. 23, no. 10, p. 4643, May 2023.

[23] P. J. Navarro, L. Miller, F. Rosique, C. Fernández-Isla, and A. Gila-Navarro, "End-to-end deep neural network architectures for speed and steering wheel angle prediction in autonomous driving," *Electronics*, vol. 10, no. 11, p. 1266, May 2021.

[24] U. M. Gidado, H. Chiroma, N. Aljojo, S. Abubakar, S. I. Popoola, and M. A. Al-Garadi, "A survey on deep learning for steering angle prediction in autonomous vehicles," *IEEE Access*, vol. 8, pp. 163797–163817, 2020.

[25] X. Lin, J. McConnell, and B. Englot, "Robust unmanned surface vehicle navigation with distributional reinforcement learning," 2023, *arXiv:2307.16240*.

[26] G. Baek, M. Saeed, and H.-K. Choi, "DuckWeeds: Their utilization, metabolites and cultivation," *Appl. Biol. Chem.*, vol. 64, no. 1, p. 73, Dec. 2021.

[27] Ultralytics. *YOLOv5*. Accessed: Jul. 5, 2023. [Online]. Available: https://github.com/ultralytics/yolov5

[28] H. Ultralytics. *YOLOv8*. Accessed: Nov. 1, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[29] PyTorch. *Models and Pre-Trained Weights*. Accessed: Aug. 14, 2023. [Online]. Available: https://pytorch.org/vision/main/models.html

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] M. Tan and Q. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10096–10106.

[32] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976.

[33] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[34] RecyclingRush. *Tests Performed at Alto Obrajes Olimpic Pool in la Paz, Bolivia*. YouTube. Accessed: Nov. 2, 2023. [Online]. Available: https://youtube.com/shorts/XEJ-U3jsgNY

[35] R. Team. *Tests Performed in Lagoon Achocalla, la Paz, Bolivia*. YouTube. Accessed: Nov. 2, 2023. [Online]. Available: https://youtu.be/sbR8at6v_68

[36] M. Aquino. *South America's Lake Titicaca Nears Record Low Water Level as el Nino Bites*. Accessed: Nov. 1, 2023. [Online]. Available: https://www.reuters.com/world/americas/south-americas-lake-titicaca-nears-record-low-water-level-el-nino-bites-2023-11-03/

EDWIN SALCEDO received the M.B.A. degree in digital businesses from Valencia Polytechnic University, Spain, in 2017, and the M.Sc. degree (Hons.) in advanced software engineering from The University of Sheffield, England, U.K., in 2019. He is a Lecturer and a Research Engineer with the Department of Mechatronics Engineering, Universidad Católica Boliviana "San Pablo." He is also an Invited Lecturer with Universidad Autonoma Gabriel Rene Moreno, Santa Cruz, Bolivia. He has coauthored articles in image processing, stereo vision, and deep learning. His research interests include pattern recognition for image processing and sensor technologies.

YAMIL UCHANI is currently a University Student with Universidad Católica Boliviana "San Pablo" and an Active Member of the Department of Mechatronics Engineering, Student Scientific Society. He has participated in various environmental projects and projects related to robotics, education, and video game development, primarily focusing on simulation and data collection. He has also collaborated on a project with the University of Louvain, Brussels. As part of a team, he won the International OpenCV AI Competition, in 2022, with the preliminary version of the project Recycling Rush.

MISAEL MAMANI (Student Member, IEEE) is currently pursuing the degree with Universidad Católica Boliviana "San Pablo." He is an Active Member of the Department of Mechatronics Engineering, Student Scientific Society. He has participated in several projects related to robotics, embedded systems, and computer vision, primarily focusing on programming. He has also collaborated on a project with the University of Louvain, Brussels. His research interests include compression techniques on deep learning for embedded devices and sensor technologies.

MARIEL FERNANDEZ is currently pursuing the degree in mechatronic engineering with Universidad Católica "San Pablo." She is an Active Member of the Student Scientific Society of Mechatronic Engineering, having participated in various multidisciplinary projects in the environmental field and projects related to robotics in the technology sector, primarily focusing on hardware and mechanics. Additionally, she has involved in a collaborative project with the University of Louvain, Brussels, Belgium. As part of a team, she won the International OpenCV AI Competition, in 2022.