Mapping Anthropogenic Ocean Litter with an Autonomous Underwater Vehicle

New Mexico Supercomputing Challenge

April 3, 2023

Team 14

Los Alamos High School

Team Members:

Daniel Kim

Teacher:

Michela Ombelli

Project Mentor:

None

Mapping Anthropogenic Ocean Litter with an Autonomous Underwater Vehicle

Daniel Kim [a*]

daniel.kim@studentlaschools.net

[a)] Los Alamos High School, Los Alamos NM

EXECUTIVE SUMMARY

This project explores the possibilities of pairing an autonomous underwater vehicle (AUV) with a deep-learning computer vision model for marine debris mapping. A cost effective, 3D-printed AUV with a motorized ballast system was designed to collect underwater footage continuously at various depths for several weeks. A simulated underwater environment using hardware-in-the-loop (HIL) procedures was used to test and evaluate the AUV. A trash detection machine learning model was developed to analyze the footage for underwater litter. To assess the accuracy and capabilities of the trash detection model, footage from various underwater vehicles was compiled and run through the model, yielding five areas of highly concentrated ocean debris at depths of 500-800 meters below the surface. This study highlights how many pieces of marine debris - undetectable by satellite data – can be mapped and categorized with the proposed AUV and trash detection model.

KEYWORDS: Autonomous underwater vehicle; Hardware-in-the-loop; Tracking and mapping; Marine debris; Real-time object detector; Machine learning.

1. INTRODUCTION

The use of plastics in the past 65 years has significantly outpaced the use of all other materials. Of the 8.3 billion metric tons of plastic ever produced, only 2500 metric tons are currently in use (Geyer et al., 2017). These plastics are the most dominant type of litter found in our oceans and contribute to the 244,000 metric tons of marine debris (Parker, 2022).

This issue has already caused substantial ecological and economic problems. An estimated 845 million people are at risk of iron, zinc, or vitamin A deficiency because of declining fish populations (Golden et al., 2016) and global ecosystem delivery systems are projected to lose $500-$2500 billion in value of benefits (Beaumont et al., 2019) because of global plastic accumulations.

Currently, quantitative assessments of marine litter are primarily done through satellite imagery. For example, the MARLISAT project combines orbital imagery and machine learning algorithms to detect plastic along beaches and oceans (Petersen, 2022). Although this method has given a clear view on concentrations of surface-level plastic, about 40% of plastics in the ocean are less dense than seawater (Andrady, 2011), leaving them undetectable through satellite imagery.

One of the largest garbage patches in the world, the Great Pacific Garbage Patch, was mapped using a fleet of 18 vessels and 642 surface nets (Lebreton et al., 2018). This method created a comprehensive understanding of the largest garbage patch in the world but required vast resources and manpower for a single assessment. The ceaseless movement of ocean debris requires continuous observation, but the use of large resources for one measurement is not sustainable. Therefore, advanced marine debris detection systems with autonomous capabilities are needed for the effective mapping of marine debris.

The purpose of this study is to design and develop an autonomous underwater vehicle (AUV) and a trash detection machine learning (ML) model to assist in marine litter quantification. The AUV is capable of surveying and capturing footage of different marine environments at various depths for long distances, while gathering other seminal metrics such as temperature and salinity data. Unlike satellites, AUVs allow for metrics to be collected at various depths, and contrary to current surface net and trawl quantification methods, AUVs are cheap and require very little manpower. The ML model analyzes the post-deployment footage from the AUV for litter, which is used to build a comprehensive map of marine debris.

Using this method, the performance and capabilities of the engineered AUV are evaluated and open-sourced underwater video datasets are analyzed to create a model of ocean litter off the coast of Kamaishi, Japan. Using the ML model, the possibility of using AUVs to collect data on the scope of marine debris was assessed.

2. METHODOLOGY

2.1 Design of AUV

The AUV is modeled after an underwater glider - a robotic vehicle suited for data collection in remote locations at a low cost. Underwater gliders do not have propellers or engines; they utilize changes in buoyancy to move up and down which creates lift to propel the vehicle forward. This method allows for the glider to operate for days, weeks, or even months autonomously before being recovered for data.

The AUV was first designed in computer-aided design (CAD), and 3D-printed. Each component of the AUV was a separate, modular design, which was then assembled onto a steel rod backbone. The entire assembly then slid into a 102 mm×914 mm clear PVC tube and was sealed with an O-ring endcap. Each of the 3D-printed modules was manufactured in polylactic acid (PLA), with high infill to increase the density of the vehicle. Since the inner printed parts are independent from the outer PVC tube, none of the inner components required water proofing. By assembling the AUV primarily with 3D printed components, the design allowed for modularity and repeatable manufacturing of parts – all at a very low cost.

The AUV was assembled using the 3D printed parts and had a width of 10.16 centimeters, a wingspan of 0.72 meters, and a length of 1.08 meters. Initially, the mass of the vehicle was insufficient to sink, thus 3.63 kg of copper plated lead weights were added to the interior of the AUV to make it neutrally buoyant, giving it a final mass of 6.63 kg.

The AUV control board, the circuit board that operates the vehicle, was first designed, and developed in electronic computer-aided design, then outsourced for fabrication by JLCPCB. Once the bare printed circuit board (PCB) was fabricated, individual integrated circuits were soldered using reflow. The microcontroller used on the control board was the Teensy® 4.1 (Teensy 4.1, 2020), featuring the Arm® Cortex-M7 (Cortex-M7, 2014) at 528 MHz and 1024 KB of memory. The software that operated the control board was developed in C++ and a complementary graphical user interface (GUI) was developed in Typescript.

2.2 Trash Detection Model

The trash detection model was built on YOLOv5, an open-source ML framework. The dataset used to train the neural network was the Trash ICRA-19 dataset (Fulton, 2019), which contained 5700 annotated images of underwater trash. Training was performed on a Google Colab server with a NVIDIA Tesla T4 with 16GB of memory and 2560 CUDA cores. A total of four different neural networks were trained and evaluated to find the best-suited model for trash detection. Table 1 summarizes the four different models. Once the models were trained, validation and deployment of the models was performed on a local server featuring the NVIDIA 1650Ti with 4GB of memory and 896 CUDA cores.

Table 1. The four models trained for trash detection. The model that showed the highest accuracy is then used for trash detection.

Model	Parameters (millions)	FLOPs @ 640 pixels (billions)
YOLOv5n	1.9	4.5
YOLOv5s	7.2	16.5
YOLOv5m	46.5	49.0
YOLOV5l	86.7	109.1

3. RESULTS AND DISCUSSION

3.1 Component Design for AUV

The AUV consists of four key components: a ballast tank to control buoyancy, a microcontroller system to control and operate the vehicle, aluminum wings to push the vehicle forward when rising and sinking, and a camera module to collect video footage of the surroundings (Figure 1).

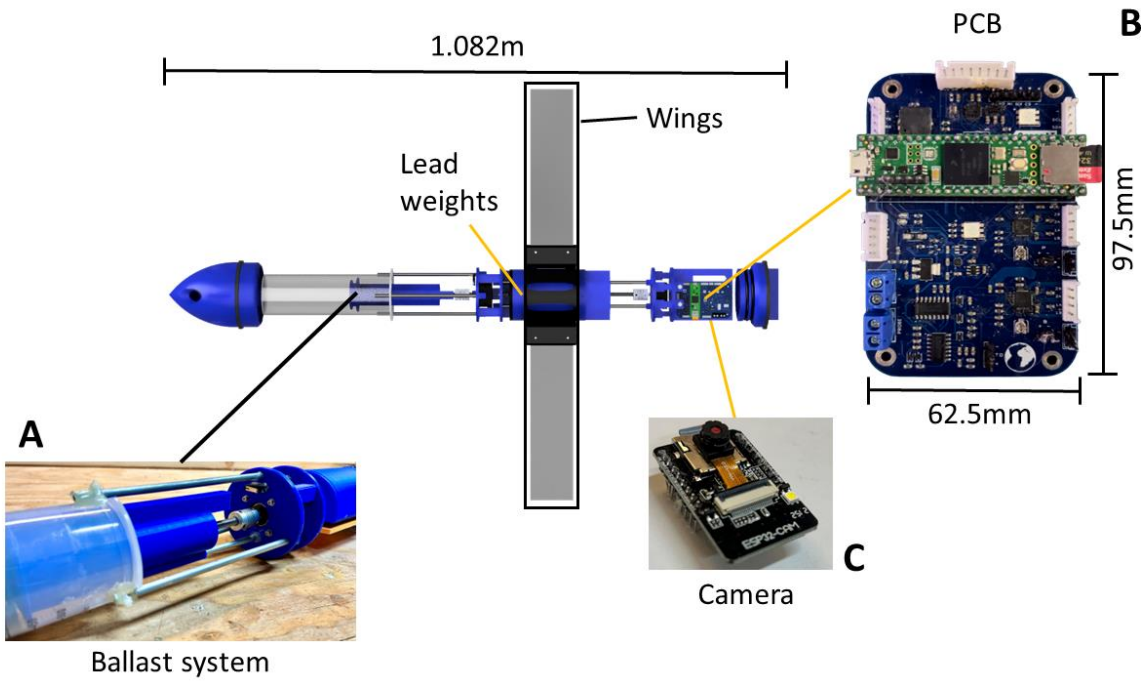

Figure 1. The AUV with its key components.

3.1.1 Ballast System

The buoyant force, F_B, exerted on the vehicle is equivalent to the weight of the volume of water displaced by the vehicle. If the buoyant force is greater than the weight of the vehicle, F_W, the vehicle floats. If the weight of the vehicle is greater than the buoyant force, the vehicle sinks. The AUV controls its weight by using a ballast tank. By pulling in water from its surroundings, the AUV increases in weight and can sink, and by pushing that water out, the AUV can then float.

For the AUV in this study, the ballast system utilizes a 550 mL syringe with a plunger driven by a stepper motor (Figure 2C). A threaded rod is connected to the stepper motor (Figure 2A) which pushes the syringe plunger in and out. The high accuracy of the stepper motor allows for fine tune adjustments to the amount of water in the tank to be controlled with very low power consumption. However, unlike servo motors, stepper motors provide no positional feedback, thus a limit switch is added to the end of the plunger's range of motion. By pulling the plunger until the limit switch is pushed (Figure 2B), the position of the plunger can be determined. The ballast assembly was able to pull in 450 mL of the 550 mL capacity, allowing for an additional 0.450 kg of water to be added to the AUV.

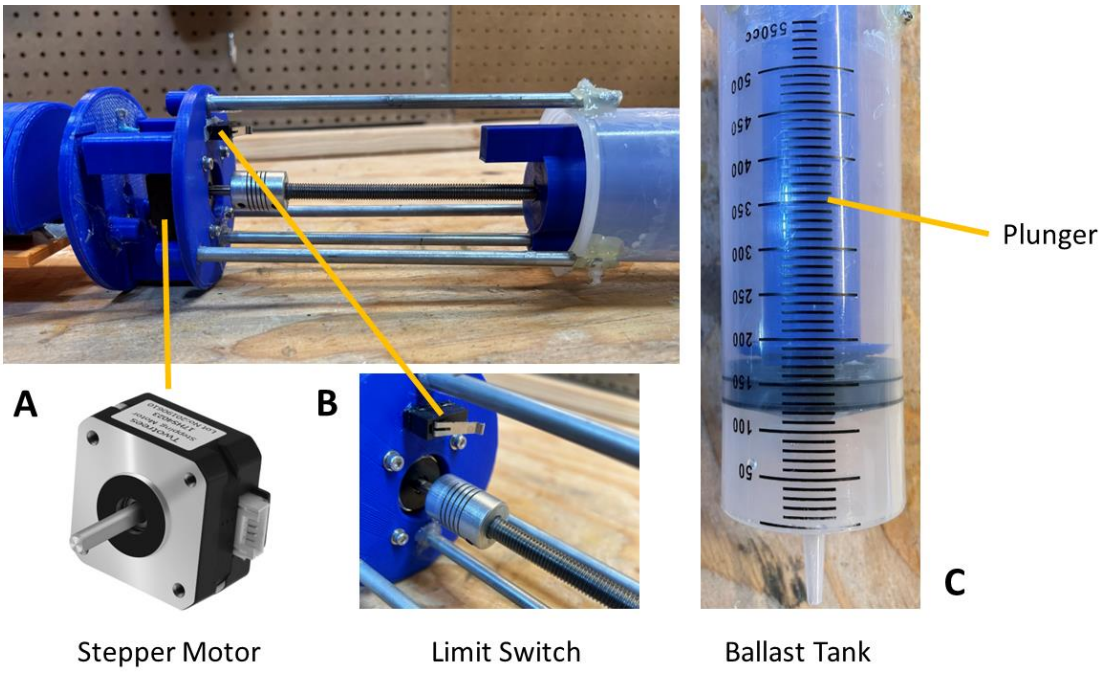

Figure 2. The AUV's ballast system.

To test the ballast system, the AUV was placed in a tub of water and was programmed to continuously fill and empty the ballast tank to sink and float. Initially, the AUV was too buoyant to sink, so an additional 308 g of lead was added to the rear of the AUV, for a final mass of 6.629 kg. Lead was chosen due to its high density and relatively low cost. The lead was copper plated to reduce the environmental and safety hazards associated with it. After the addition of the extra weight, the AUV was able to control its buoyancy with the ballast system and repeatedly float and sink (Figure 3).

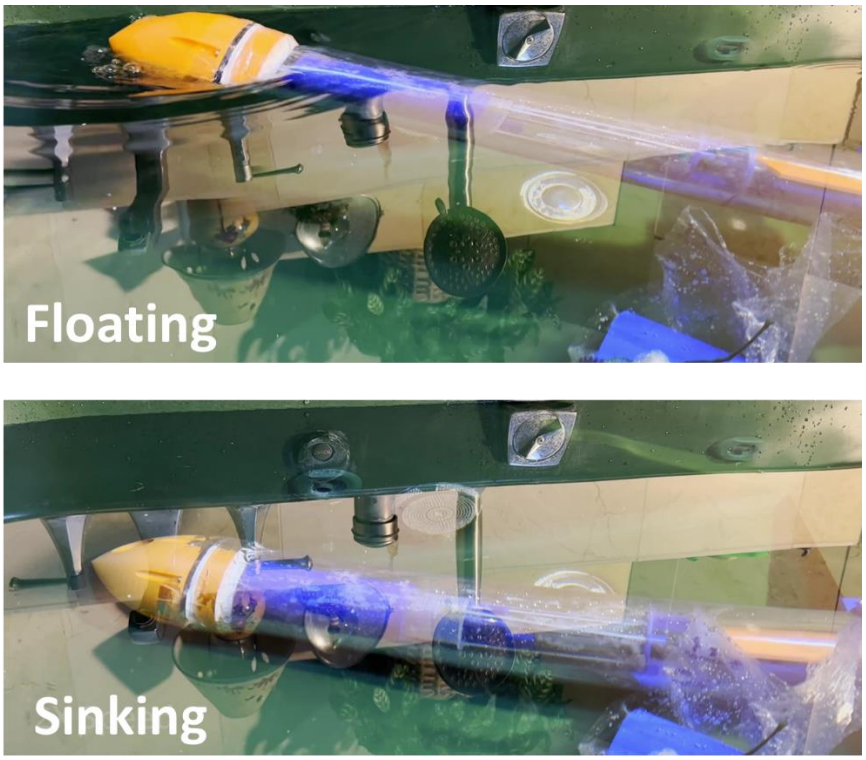

Figure 3. Field test of the AUV's ballast system.

3.1.2 AUV Control System

To read sensor data, control and navigate the AUV, and perform telemetry, a custom printed circuit board was designed and assembled (Figure 1B). At its core, the control system has a 528 MHz microcontroller unit that runs the AUV control software and controls all aspects of the vehicle.

To calculate the orientation and relative speed and position of the vehicle, the control board features an accelerometer, gyroscope, and magnetometer.

The board also includes a total dissolved solids (TDS) sensor to calculate external solute concentrations. The TDS sensor calculates the concentration of solutes within the AUV's environment by measuring the current across two electrodes providing 3 V AC power at 50 Hz. The excitation source is alternating instead of direct current to prevent the sensor from polarization by preventing the buildup of charged particles on the electrodes. Since the conductivity of water increases as temperature increases, a thermistor is installed next to the TDS probe to calculate a more accurate reading.

To collect data needed for marine litter mapping, a GPS and external pressure sensor are all connected to the board. These two sensors allow for the location and depth of any detected litter to be identified. All data collected from these sensors are then logged onto a microSD

card. The ballast tank is controlled through a stepper motor, which the control system operates using a stepper motor driver.

Since most radio frequencies greater than 1 MHz do not work at distances greater than 10 meters underwater (Qureshi, 2016), the control board allows for wired connections through the serial peripheral interface (SPI) or inter-integrated circuit (I2C) protocols. Although not implemented, connections to external modules on the surface (using tethered buoys) would allow for telemetry and GPS data to be transmitted and received. The SPI and I2C interfaces also allow for other additional components to be added to the AUV framework, such as auxiliary sensors or cameras.

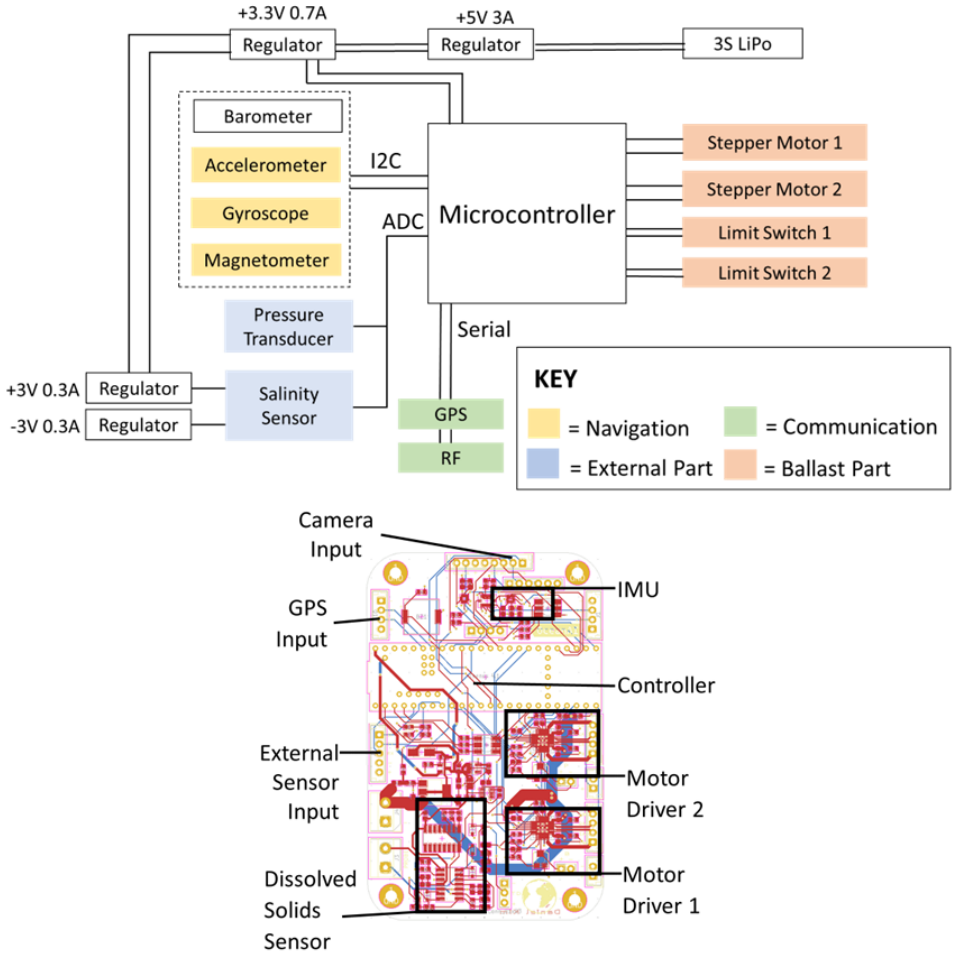

Figure 4. The components of the control board (top) and the labeled ECAD model of the control board (bottom)

3.1.3 Camera System

Although the main control board allows for an external camera module to be connected, image processing and logging are very CPU and memory intensive operations, thus a separate module was added to collect footage of the AUV's surroundings. The ESP32-S microcontroller with the OV2640 2-megapixel camera was chosen to collect footage of the AUV's surroundings (Figure

1C). The 32-bit ESP32 microcontroller runs at 240 MHz and features 2 cores and 520 KB of RAM, which is sufficient to capture video footage continuously and save to a microSD card. The small size of the module (27mm × 49mm) allows for multiple camera units to be setup within the AUV, which can provide multiple camera angles.

3.1.4 AUV Control Software

Although the control system and camera system provided the hardware to operate the AUV, respective software had to be developed to run the control system. The software controls everything from the stepper motors to drive the ballast tank to the data collection and logging. The AUV control software is written in C++ and had two key aspects: sensor data collection and data logging/telemetry.

3.1.4.1 Sensor Data Collection

The AUV had seven key sensors that collected data about the state of the vehicle. The TDS sensor measures the number of solutes in the environment in conjunction with an external temperature and pressure sensor, while an accelerometer, gyroscope, magnetometer, and GPS were used to measure orientation and provide an estimate of the AUV's position.

To calculate the external solute concentration with the TDS sensor based on temperature, the temperature compensated reading, ϕ_T must be first calculated:

$$\phi_T = \left(\frac{A_S * V_{REF}}{1023}\right) 1.0 + 0.02(T - 25) \quad (1)$$

where:

- A is the raw analog reading from the TDS sensor.
- V_{REF} is the reference voltage of the microcontroller.
- T is the temperature in degrees Celsius measured by the thermistor.

This is necessary because the conductivity increases as temperature increases. The solute concentration can then be calculated by

$$S_T = 0.5 * [133.42 * {\phi_T}^3 - 255.86 * {\phi_T}^2 + 857.39 * \phi_T] \quad (2)$$

where:

- S_T is the solute concentration in g/mL at temperature T

Temperature readings from the thermistor not only help provide a more accurate TDS measurement but can also be used to measure rising ocean temperatures throughout multiple AUV deployments. The temperature is first calculated by measuring the resistance of the thermistor by calculating the voltage drop:

$$R_T = (V_{REF} - V_{therm}) * R_k \div V_{therm} \quad (3)$$

where:

- R_T is the resistance of the thermistor in ohms at temperature T.
- V_{REF} is the reference voltage of the microcontroller.
- V_{therm} is the voltage measured across the resistor.
- R_k is the known resistance in the voltage divider.

The temperature can then be calculated with the Steinhart-Hart equation as:

$$T = A_c + B_c \ln(R_T) + C_c[\ln R_T]^3 - 273.15 \quad (4)$$

where:

- A_c, B_c, and C_c are the Steinhart-Hart coefficients varying on the type of thermistor used.
- T is the temperature measured in degrees Celsius.

External pressure readings provide accurate measurements of the depth of the vehicle. The pressure transducer outputs a simple analog output directly proportional to the pressure of the environment. The accelerometer, gyroscope, and magnetometer, (collectively referred to as the inertial measurement unit or IMU here) are each separate devices that provide readings to the microcontroller through the I2C communication protocol. For positioning, the AUV utilizes GPS data for absolute positioning and the IMU for localization. The GPS module provides horizontal positioning up to 2.5 m in accuracy at 10Hz while the IMU provides accurate horizontal positioning through dead reckoning. Using GPS, pressure measurements, and the IMU, the position of the AUV in all three dimensions can be estimated. The IMU is also used to calculate the orientation of the vehicle using the Madgwick algorithm (Madgwick, 2010). The Madgwick algorithm fuses accelerometer, gyroscope, and magnetometer readings to provide a low-drift estimation of the AUV's orientation.

3.1.4.2 Data logging and Telemetry

During deployment, data is serialized onto a microSD card in the JSON format. A converter was also developed in C++ to convert the JSON data into more easily readable formats, such as CSV. Data is transmitted to an external GUI through the universal asynchronous receiver-transmitter (UART) protocol. The GUI is built in Typescript-React and provides live data feeds from the sensors and allowed for control over the AUV's ballast system. The GUI allows for extensive control of the vehicle's ballast system and provided critical information such as vehicle orientation, external pressure, and battery voltage. Telemetry could be sent wirelessly through a tethered buoy but is sent directly through a wire directly connected to the control board during AUV testing.

3.2 Hardware-In-The-Loop

After designing and assembling the AUV and its respective software, the platform required extensive testing to evaluate its effectiveness. Unfortunately, many garbage patches exist in the

middle of the ocean and the means of testing this application is geographically restricted. To overcome this challenge, a method called hardware-in-the-loop (HIL) was adopted to test the AUV. HIL testing simulates reality by feeding real signals through the platform while test and design iteration take place. HIL allows for comprehensive testing of complex systems through many of possible scenarios without spending time and money associated with real-world tests (NI, 2022).

Data from the open-sourced Slocum glider from Teledyne Webb Research was fed through the software while simulating underwater conditions (Figure 5). The data from this dataset included 1.139 megabytes of latitude, longitude, depth, pressure, temperature, and salinity data collected from May 25, 2018, to July 16, 2019. Although the AUV continuously read from the onboard sensors, the readings from the sensors were overridden with data from the dataset. While the data was fed through the system, the AUV performed its normal operations, such as ballast tank control, telemetry, sensor data reading and logging, and underwater footage collection. These tests evaluated the functionality of the sensors and mechanical aspects of the AUV, while also testing the abilities of the software and control board to operate the AUV.

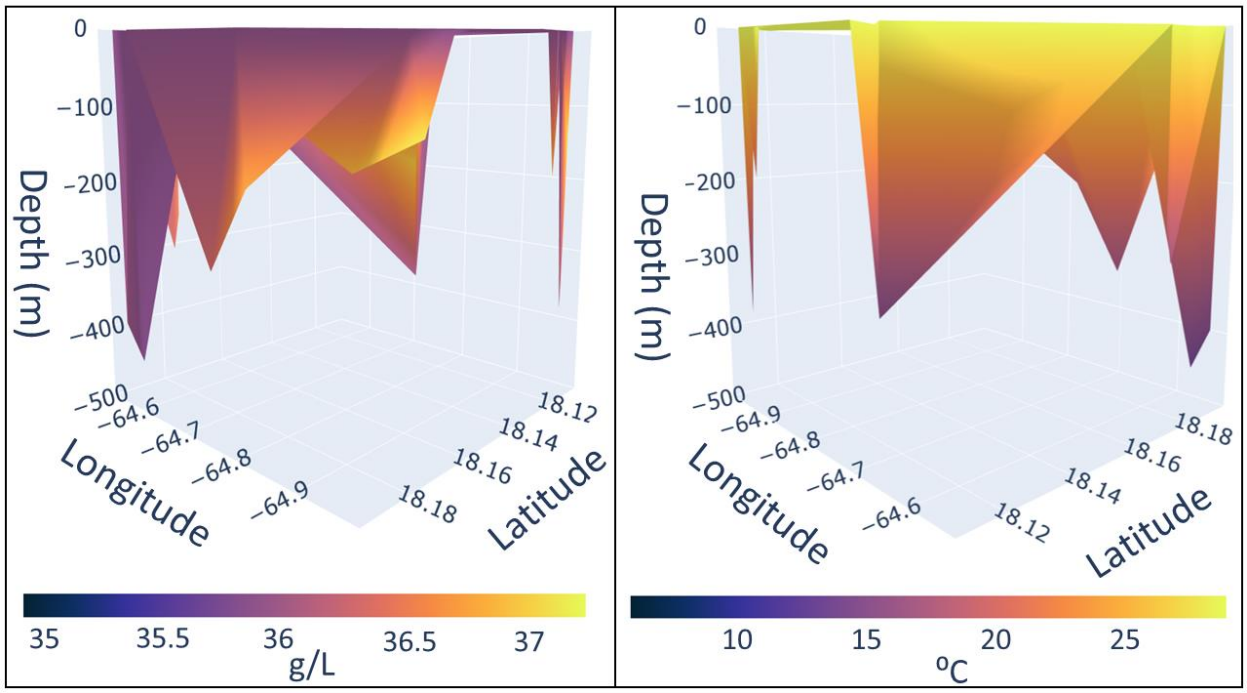

Figure 5. Salinity data from the HIL test in grams per liter (left) and the temperature data from the HIL test in degrees Celsius.

3.3 Trash Detection Model

The trash detection model locates and classifies litter in an image or video using a deep neural network. For each image or frame of a video, the model outputs a predicted bounding box surrounding the piece of litter. Once the AUV is recovered, the footage and sensor data are run through this model. If a piece of litter is detected, the model then extracts the respective depth

and GPS data, which can be used to build a comprehensive map of ocean litter. The trash detection model trained in this study only has one class – meaning that it can detect if there is trash or no trash within a certain image. Training was done on the Trash ICRA-19 Dataset (Fulton, 2019) which contained 5700 images of underwater trash, each of which were annotated with a ground truth bounding box which enclosed the litter within the image.

3.3.1 Training

Before the model could be deployed, it needed to first be trained on a large dataset. For training, four different convolutional neural networks (CNNs) were used: YOLOv5n, YOLOv5s, YOLOv5m, and YOLOv5l (Jocher, 2023). To evaluate the best model to use for trash detection, several metrics were measured throughout the model training. The model that exhibited the best metrics would be chosen as the final detection model.

In machine learning, true positives, TP, are predictions that the model made correctly. False positives, FP, are outcomes where the model incorrectly predicted that trash was detected. False negatives, FN, on the other hand, are instances where trash was present, but the model did not predict litter at that position.

Using these metrics, the precision and recall of the model can be calculated. Precision is defined as:

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

Precision gives the proportion of positive identifications that were correct. By contrast, recall gives the proportion of actual positives that were identified correctly, and is defined as:

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

For each of the images the model inferences, it returns a confidence score which shows the probability of trash being within the bounding box. If the confidence score is greater than the confidence threshold, a value set by the user, the object is classified as trash. Thus, the confidence threshold affects the ratio of true positives, false positives, and false negatives, and therefore causes a tradeoff between precision and recall. Typically, a high-precision model will have a lower recall and vice versa. Therefore, to evaluate the model, a different metric, called mean average precision, mAP, must be used. mAP accounts for both precision, recall, and differing confidence thresholds giving an intuitive measure of the accuracy of a model. To calculate mean average precision, first, the average precision, AP, for each class of the model is calculated. AP is defined as:

$$\int_0^1 p(r)dr \quad (7)$$

where:

- $p(r)$ is the precision and recall function, giving us the precision of the model at a specific recall, r

Thus, a model that can detect 5 different categories (classes) will have 5 AP values. To calculate the mean average precision, the mean of the AP values is calculated.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (8)$$

where:

- AP_k is the average precision of class k.
- n is the number of classes.

Since the trash detection model only has one class, the mean average precision of the model is equivalent to the average precision. Although mean average precision gives an overall quantification of the performance of the model based on precision and recall, another metric, loss, is also needed to measure the accuracy of the bounding boxes produced by the model. Loss consists of three parts: classification loss, localization loss and objectness loss (Liu, 2016). Localization loss measures differences between the predicted and ground truth bounding box coordinates, while confidence loss measures the probability of object presence or absence within a predicted bounding box. Classification loss is a measure of the model's ability to predict the correct class. However, since the trash detection model only has a single class, this metric is negligible. Localization loss, L, is defined as:

$$\gamma_c \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{I}_{ij}^{obj}[(x_i + \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \gamma_c \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{I}_{ij}^{obj}[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2] \quad (9)$$

where:

- γ_c is the scaling factor for localization loss.
- S^2 is the number of grid cells in the output feature map (the output of the convolutional layers).
- B is the number of bounding boxes predicted for every grid cell.
- $\mathbb{I}_{ij}^{obj}$ is the indicator variable that equals 0 unless the j-th bounding box in the i-th grid cell is responsible for detecting an object, in which case it equals 1.
- x_i and y_i are the coordinates of the center of the predicted bounding box.
- w_i and h_i represent the width and height of the predicted bounding box, respectively.
- $\hat{x}_i$ and $\hat{y}_i$ are the coordinates of the center of the ground-truth bounding box.

- $\hat{w}_i$ and $\hat{h}_i$ represent the width and height of the ground-truth bounding box, respectively.

If an object is detected within the box, the objectness loss O is:

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \, (C_i - \hat{C}_i)^2 \tag{10}$$

If an object is not detected within the box, O is:

$$\gamma_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} \, (C_i - \hat{C}_i)^2 \tag{11}$$

where:

- S^2 is the number of grid cells in the output feature map (the output of the convolutional layers).
- B is the number of bounding boxes predicted for every grid cell.
- $\mathbb{I}_{ij}^{obj}$ is the indicator variable that equals 0 unless the j-th bounding box in the i-th grid cell is responsible for detecting an object, in which case it equals 1.
- $\mathbb{I}_{ij}^{noobj}$ is the complement to $\mathbb{I}_{ij}^{obj}$.
- C_i is the predicted objectness score for the j-th bounding box in the i-th grid cell.
- $\hat{C}_i$ is the ground-truth objectness label for the j-th bounding box in the i-th grid cell.
- γ_{noobj} is a scaling factor that controls the weight of the no object loss when detecting background.

The final loss of the single class model is the sum of the localization and objectness loss defined as:

$$\gamma_c \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} [(x_i + \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \gamma_c \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} [\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \, (C_i - \hat{C}_i)^2 + \gamma_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} \, (C_i - \hat{C}_i)^2 \tag{12}$$

The loss function was calculated on the training images and on a separate validation dataset that was not used during training. By comparing the training loss with the validation loss, the model can be evaluated for overfitting. In general, if the validation loss is greater than the training loss, the model is overfitting because it has learned to predict only the training data too closely.

In short, mean average precision measures the ability of a ML model to detect objects and loss measures the model's ability to minimize the difference between ground-truth and its predictions. Figure 6 and Table 2 show the different metrics of each of the four CNNs that were trained throughout the 200 epochs for which the model was trained. The YOLOv5l model exhibited the highest mean average precision and the lowest training and validation loss, while not showing significant signs of overfitting.

Table 2. Best and average mean average precision, training loss, and validation loss for each neural network model. The percent difference defines the difference between the best training and validation loss metric.

	YOLOv5s	**YOLOv5n**	**YOLOv5m**	**YOLOv5l**
mAP (best)	0.706	0.720	0.759	0.774
mAP (average)	0.548	0.554	0.612	0.616
Training loss (best)	0.105	0.105	0.105	0.105
Training loss (average)	0.059	0.059	0.056	0.056
Validation loss (best)	0.105	0.105	0.105	0.105
Validation loss (average)	0.059	0.059	0.056	0.056
Percent difference (best training vs. best validation)	0.000	0.000	0.002	0.000

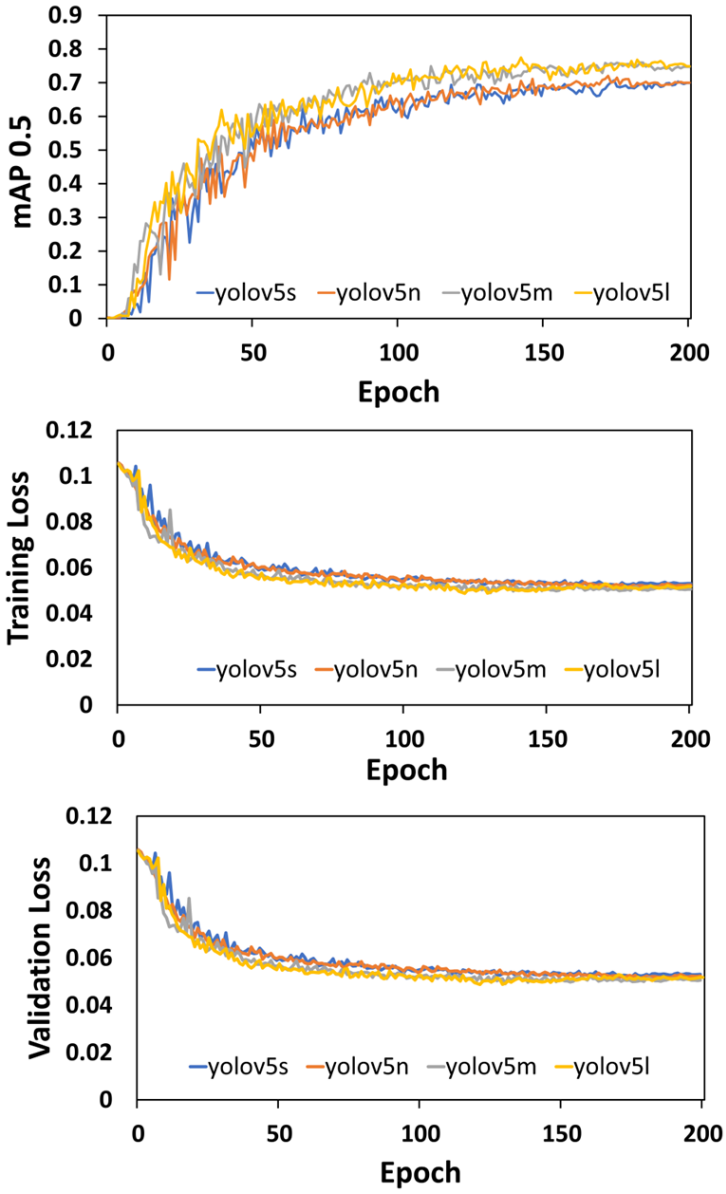

Figure 6. Mean average precision at intersection over union threshold 0.5 throughout the 200 epochs each model was trained on (top). Training loss of each model throughout 200 epochs (middle), and the validation loss of each model throughout 200 epochs (bottom). As the model trains for more epochs, it slowly improves by increasing the mean average precision and decreasing the loss.

3.3.2 Trash Detection Model Deployment

Since the AUV was tested using hardware-in-the-loop, it produced no real footage to run through the trash detection model. To evaluate the abilities of the model, open-source footage was collected from the Japan Agency of Marine-Earth Science and Technology (JAMSTEC)

HYPER-DOLPHIN submersible. The model detected a total of 174,734 litter objects at depths of 500-800 m below the surface. Objects were considered trash if the confidence score was greater than the confidence threshold, set at 0.4. Figure 7 shows the model inferencing a frame from the submersible and the detected objects from the model deployment.

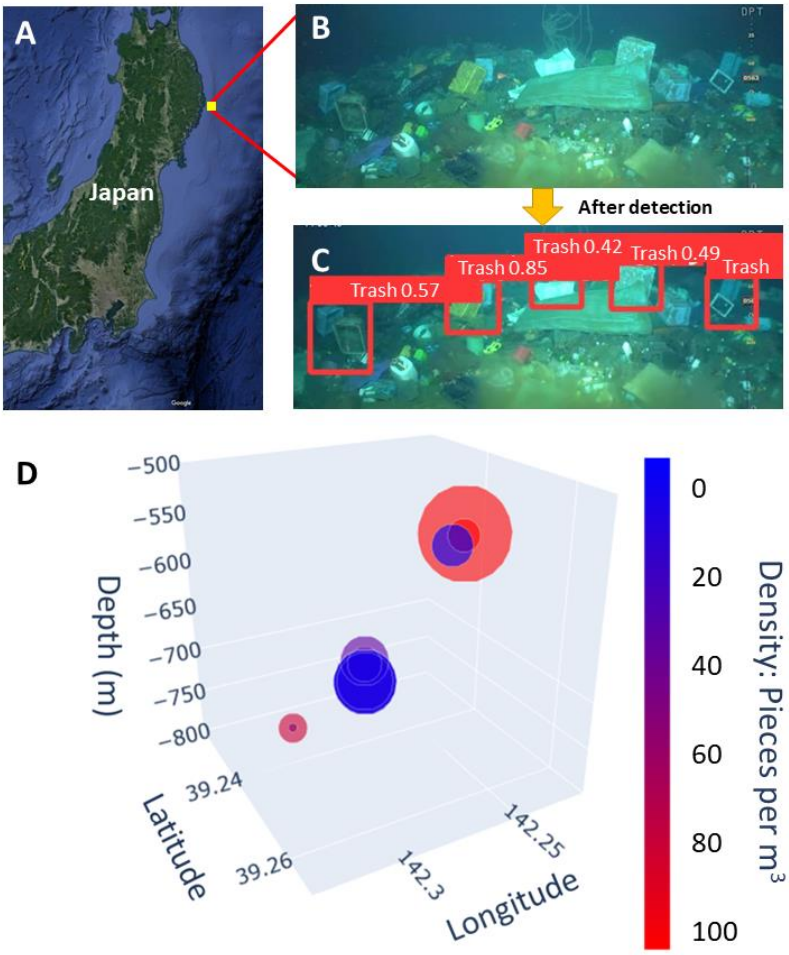

Figure 7. A: Location of the footage collected from the submersible off the coast of Kamaishi. B: Raw footage from the submersible. C: Inferenced image after being processed by the YOLOv5l trash detection model. D: Areas of debris detected by the model and the relative density of litter in each area represented by color.

4. CONCLUSIONS

This study shows an engineered AUV along with the training and deployment of a trash detection model. As shown by the HIL tests, the AUV can accurately read sensor data and process footage while navigating underwater, and the high precision of the computer vision model can seamlessly detect and analyze litter within underwater footage. The development of both a prototype AUV and novel trash detection computer model indicate that this system would be a viable method to quantify and map litter concentrations across our world's seas.

The AUV demonstrated an ability to collect sensor data and operate in underwater conditions, while the trash detection model found 174,000 debris objects underwater at depths undetectable to satellites and surface net analysis.

An exploration into replacements to the external parts of the AUV (the outer shell of the vehicle in contact with the environment) could yield better materials to handle salty ocean conditions and the use of more durable plastics for 3D printed parts could give the vehicle a more robust design. However, the biggest improvement is for more testing and evaluation to be done on the vehicle. Further tests in local bodies of water or even in the ocean could yield important flaws within the vehicle design. Furthermore, I hope to implement usage of the accelerometer onboard the AUV to track the forces of ocean currents to track the movement of debris, rather than just the position.

Currently, the trash detection model has only one classification level – whether trash is in the image or not. By using a more comprehensive dataset during training, the model would be able to classify different types of debris (e.g., plastic, metal, glass). Furthermore, by tuning hyperparameters within the existing model (number of epochs, learning rate, and model architecture), the mean average precision and loss of the model can be improved.

AUTHOR INFORMATION

Corresponding Author
Daniel Kim
Los Alamos High School
1300 Diamond Drive
Los Alamos, New Mexico 87544
Email: daniel.kim@studentlaschools.net

APPENDIX A

Symbol	Description
AUV	Autonomous Underwater Vehicle
CAD	Computer-aided-design
CNN	Convolutional Neural Network
ECAD	Electrical computer-aided-design
GUI	Graphical user interface
HITL	Hardware-in-the-loop
I2C	Inter-integrated circuit
IMU	Inertial measurement unit
ML	Machine Learning
PLA	Polylactic acid
PVC	Polyvinyl chloride
SPI	Serial peripheral interface
TDS	Total dissolved solids
UART	Universal asynchronous receiver-transmitter
A_c	Steinhart-Hart coefficient
AP	Average precision
AP_k	Average precision of class k
A_s	Analog reading
B	Number of bounding boxes predicted for every grid cell
B_c	Steinhart-Hart coefficient
C_c	Steinhart-Hart coefficient
C_i	Predicted objectness score for the j-th bounding box in the i-th grid cell
$\hat{C}_i$	Ground-truth objectness score for the j-th bounding box in the i-th grid cell
F_B	Buoyant force
FN	False negative
FP	False positive
F_W	Weight force
h_i	Height of the predicted bounding box
$\hat{h}_i$	Height of the ground-truth bounding box
$\mathbb{I}_{ij}^{obj}$	Indicator variable that equals 0 unless the j-th bounding box in the i-th grid is responsible for detecting an object, in which case it equals 1
$\mathbb{I}_{ij}^{noobj}$	Complement to $\mathbb{I}_{ij}^{obj}$
mAP	Mean average precision
n	Number of classes
$p(r)$	Precision value at recall level r
R_k	Known resistance in the voltage divider
R_T	Resistance at temperature T
S_T	Solute concentration in g/mL at temperature T
S^2	Number of grid cells in the output feature map

T	Temperature reading in degrees Celsius
TP	True Positive
V_{REF}	Microcontroller reference voltage
V_{therm}	Voltage measured across the thermistor
w_i	Width of the predicted bounding box
$\hat{w}_i$	Width of the ground-truth bounding box
x_i	X coordinate to the center of the predicted bounding box
$\hat{x}_i$	X coordinate to the center of the ground-truth bounding box
y_i	Y coordinate to the center of the predicted bounding box
$\hat{y}_i$	Y coordinate to the center of the ground-truth bounding box
ϕ_T	Temperature compensated TDS reading at temperature T
γ_c	Scaling factor for localization loss
γ_{noobj}	Scaling factor that controls the weight of the no-object loss

ACKNOWLEDGEMENT

I would like to give thanks to Dr. Robert Hermes for his help in acquiring lead weights used in this study. Thank you to my parents for overseeing the many tests and evaluations required in this study.

REFERENCES

Andrady, A. L. Microplastics in the marine environment. *Mar. Pollut. Bull.* [Online] 2011, *62* (8), 1596-1605. ScienceDirect. https://www.sciencedirect.com/science/article/pii/S0025326X11003055?via%3Dihub (accessed Feb 15, 2023).

Beaumont, N. J.; Aanesen, M.; Austen, M. C.; Börger, T. et al. Global ecological, social and economic impacts of marine plastic. *Mar. Pollut. Bull.* [Online] 2019, *142,* 189-195. ScienceDirect. https://www.sciencedirect.com/science/article/pii/S0025326X19302061?via%3Dihub (accessed Feb 15, 2023).

Cortex-M7. https://developer.arm.com/Processors/Cortex-M7 (accessed Feb 21, 2023).

Fulton, M.; Hong, J.; Islam, M.; Sattar, J. Robotic Detection of Marine Litter Using Deep Visual Detection Models. *IEEE* [Online] 2019, 5752-5758. https://ieeexplore.ieee.org/abstract/document/8793975 (accessed Feb 15, 2023).

Geyer, R.; Jambeck, J.; Law, K. Production, use, and fate of all plastics ever made. Science. [Online] 2017, 3 (7), https://www.science.org/doi/10.1126/sciadv.1700782 (accessed Feb 15, 2023).

Golden, C. D.; Allison, E. H.; Cheung, W. W.; Dey, M. M. et al. Nutrition: Fall in fish catch threatens human health. *Nature* [Online] 2016, 317-320. https://doi.org/10.1038/534317a (accessed Feb 15, 2023).

Jocher, G. *YOLOv5*. https://github.com/ultralytics/yolov5 (accessed Jul 20, 2022).

Lebreton, L.; Slat, B.; Ferrari, B.; Sainte-Rose, B, et al. Evidence that the Great Pacific Garbage Patch is rapidly accumulating plastic. *Sci. Rep.* [Online] 2018, *8*, https://doi.org/10.1038/s41598-018-22939-w (accessed Feb 15, 2023).

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C, et al. SSD: Single Shot MultiBox Detector. *LNIP*. [Online] 2016, *9905,* Springer. https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2#chapter-info (accessed Feb 15, 2023).

Madgwick, S. O.; Harrison; A. J.; Vaidyanathan, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. IEEE [Online] 2011, 1-7. https://ieeexplore.ieee.org/document/5975346 (accessed Feb 15, 2023).

National Instruments. What Is Hardware-in-the-Loop?, 2002. https://www.ni.com/en/solutions/transportation/hardware-in-the-loop/what-is-hardware-in-the-loop-.html (accessed Jan 23, 2023).

Parker, L. Ocean Trash: 5.25 Trillion Pieces and Counting, but Big Questions Remain, National Geographic. https://education.nationalgeographic.org/resource/ocean-trash-525-trillion-pieces-and-counting-big-questions-remain (accessed Jan 23, 2023).

Petersen, C. C. Satellites are Tracking Rivers of Garbage Flowing Across the Oceans, 2022. Universe Today. https://www.universetoday.com/157156/satellites-are-tracking-rivers-of-garbage-flowing-across-the-oceans/ (accessed Jan 23, 2023).

Qureshi, U. M.; Shaikh, F. K.; Aziz, Z.; Shah, S, et al. RF Path and Absorption Loss Estimation for Underwater Wireless Sensor Networks in Different Water Environments. Sensors [Online] 2016, 16 (6), 890. National Library of Medicine. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4934316/ (accessed Feb 15, 2023).

Teensy® 4.1 Development Board. https://www.pjrc.com/store/teensy41.html (accessed Feb 21, 2023).