Autonomous Capture of Submerged Plastic Bags

Nicholas Guimond

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science

University of Washington
2025

Committee:
Aaron Marburg
Howard Chizeck
Ashis Banerjee
Krithika Manohar

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Autonomous Capture of Submerged Plastic Bags

Nicholas Guimond

Chair of the Supervisory Committee:

Aaron Marburg
Electrical and Computer Engineering

Plastic bags are a major contributor to oceanic macroplastic pollution, presenting serious threats to marine life and ecosystems. As these bags break down into microplastics, the hazards worsen, affecting both marine organisms and human health. This thesis introduces a complete pipeline designed to capture drifting plastic bags using an underwater manipulator. This pipeline could be used for future deployment on autonomous underwater vehicles (AUVs). The system integrates object detection, tracking, and intercept planning to capture drifting underwater objects. Chapter 1 introduces FindingTrash, a vision pipeline combining YOLO and DeepSORT, trained on a curated dataset to detect and track plastic bags. This system is paired with a robotic manipulator to enable reactive grasping. Chapter 2 builds on this by integrating a reachability-based planning strategy that compares predicted drift times with precomputed arm reach times, allowing the robot to select timely interception points. Bag drift was simplified to one degree of freedom to reduce complexity. A simulation of our system tracked and intercepted plastic bags through our test tank, offering a promising solution for automated underwater plastic bag removal.

Acknowledgements

This work was funded by the Office of Naval Research under grants N00014-21-1-2052 and N00014-22-1-2196. We also thank the Applied Physics Lab at the University of Washington for granting us access to their facilities, contributing to the success of our project.

DEDICATION

To my mother, who showed me it was possible, and to my father, who helped me along the way.

Contents

List of Figures

List of Tables

Introduction

Brief Overview

This thesis explores methods for combining object detection and motion planning of an underwater robotic arm for the autonomous retrieval of drifting plastic bags. This work is built upon recent research developed in computer vision and autonomous grasping. It focuses on the practical applications of several theoretical approaches. The thesis is structured into two main chapters. Each chapter can stand alone, but also contributes to a broader vision of integration.

Background

Plastic bags are a form of macroplastic pollution, found even in remote ocean depths. Approximately 80% of the plastic waste found in the ocean originates from land-based sources, such as discarded plastic bags [1]. This pollution poses serious risks to wildlife. Numerous marine species, such as whales, seals, and manatees, have been documented ingesting plastic. Plastic bags can resemble prey in shape, color, and motion in the water column. This behavior can lead to internal injuries, infections, digestive blockages, and even death [2]. Furthermore, ingestion of plastic bags has been shown to cause cetaceans to struggle to swim, which heightens their risk of ship strikes [3]. This suggests that plastic bags may contribute to more than just reported direct fatalities from airway obstructions or perforations.

In addition to their macroplastic threat, plastic bags pose risks to aquatic life and human health when they break down into microplastics [4]. Plastic bags are made of polyethylene and degrade over time in marine environments. Exposure to sunlight, wave action, and other environmental factors cause the plastics to fragment into microplastics that are less than 5mm in size and are even more challenging to remove from the environment [5]. Additionally, this degradation of polyethylene can lead to the release of greenhouse gases like methane and ethylene, contributing to climate change. These small particles are difficult to track and remove. Microplastics can accumulate in food webs and harm marine life. Humans may ingest these particles through contaminated seafood. This exposure is linked to respiratory and reproductive issues, with potential connections to cancer [6].

Even before degradation occurs, plastic bags may leach. Leaching of chemical additives from plastic has been shown to negatively impact aquatic environments due to their potential toxicity [7]. In addition to their immediate threats, marine pollutants can cause noticeable economic loss with negative effects on tourism, fisheries, and aquaculture [8].

Oceanic microplastic pollution is worsening, with estimates exceeding 170 trillion plastic particles in the oceans and a growth rate that continues to outpace current mitigation efforts [9]. Projections indicate a potential 50% increase in oceanic macroplastics over the next decade without intervention [3]. This is why capturing plastic bags before they degrade into microplastics is essential. Moreover, removing intact plastic bags is generally less resource-intensive than removing the equivalent amount of microplastics.

Challenges

This work faces several key challenges. First, detecting plastic bags underwater is difficult because they tend to drift with currents, change shape, and often blend into the environment. Over time, these bags degrade, becoming less visually distinct, which further complicates reliable detection. Additionally, underwater imaging is hindered by light attenuation and turbidity, making visual detection even more challenging.

Second, capturing drifting plastic bags is a complex problem due to their unpredictable motion within the water column. Planning successful grasps is made more difficult by the physical limitations of robotic manipulators, delays in perception and control systems, and the narrow time window available to intercept and grasp the object before it moves out of reach. The deformable and irregular shape of plastic bags adds yet another layer of difficulty to reliable grasping.

Our Approach

This work brings together established techniques in underwater robotics to create a system for detecting and retrieving floating plastic bags. The pipeline encompasses object detection, motion prediction with grasp planning for real-time manipulation within a Robot Operating System (ROS1)-based architecture [10].

At the core of the system is FindingPlastic, a custom-trained detection model optimized for identifying plastic bags in underwater environments. Detections are tracked over time using a multi-object tracking algorithm, improving robustness to occlusions and detection dropouts. Using synchronized stereo depth data, 2D image coordinates are projected into 3D space to estimate the position of each object. These 3D centroids inform grasp trajectories for a 7-degree-of-freedom underwater manipulator.

To simplify the challenge of dynamic plastic bag drift, we assume the bag moves only vertically, reducing the problem to a single degree of freedom. The system predicts short-term object motion based on observed drift and selects grasp poses accordingly, allowing the arm to intercept moving targets. This framework was tested in a simulation and serves as a foundation for more advanced deployment on ROVs or AUVs.

Beyond the scope of this project, grasping drifting objects underwater can be applied beyond plastic bags. The methods proposed in this research could be adapted to sample jellyfish or other marine organisms. They could also be utilized to recover items like damaged equipment and debris. These kinds of objects often have fragile components and potentially unpredictable movements. Our methods can be used to achieve safe and dependable autonomous retrieval of a variety of underwater objects.

Outline

This thesis is organized into two main chapters. Chapter 1 introduces FindingPlastic, presenting the development and evaluation of an underwater plastic bag detection and retrieval pipeline tested in a controlled tank environment. It covers detection, tracking, acquisition, and storage stages in detail.

Chapter 2 builds on this foundation by integrating the tracking system with a grasp planning framework designed to predict and intercept drifting plastic bags in real time. This chapter includes methods for motion prediction, grasp planning, and experimental evaluation of the integrated system's performance.

Finally, the work concludes by summarizing the key contributions and outlining opportunities for future work to advance autonomous underwater plastic bag remediation.

Chapter 1

FindingPlastic: Underwater Plastic Bag Detection and Retrieval

This chapter demonstrates an end-to-end pipeline for capturing floating plastic bags using a 7-degree-of-freedom manipulator in a test tank. It serves as a proof of concept for future designs in which an AUV or ROV could employ a manipulator for plastic capture. We present the complete pipeline before detailing its four stages: detection, tracking, acquisition, and storage.

1.1 Related Work

1.1.1 Underwater Pollution Object Detection

This work builds on prior projects to visually identify underwater trash. Fulton and Hong [11] use the J-EDI dataset to train a YOLOv2 (You Only Look Once) object detection model [12] for marine litter detection. Their approach categorizes marine litter into plastic, ROV, and bio categories. They take advantage of the regularly updated J-EDI dataset to improve their detection model by incorporating more data.

AquaVision, which uses deep transfer learning to automate waste identification in marine environments, was created by Panwar et al. [13]. AquaVision was trained on the Aquatrash dataset derived from the TACO dataset [14]. AquaVision demonstrated remarkable performance in detecting plastic, including plastic bags, and proved to be valuable for a range of types of marine debris.

Tata and colleagues [15] utilized the DeepTrash [16] dataset for training. The dataset is made up of 3200 images from JAMSTEC and their own collected data. The reported results show high performance with YOLOv5 [17] and YOLOv4 [18].

We have chosen to build upon the DeepTrash dataset due to its large sample size and its incorporation of plastic bags. The YOLO-based approaches validated our object detection strategy. YOLO models are convolutional neural networks that combine classification and bounding box prediction using a grid-based approach for fast and accurate object detection. Although DeepPlastic's YOLOv5 model performed well for our needs, reported observations of YOLOv8's enhanced throughput and hardware efficiency, particularly with higher image resolutions, suggested YOLOv8 would better suit our detection task [19]. Additionally, the availability of annotated dataset and pre-trained weights from DeepPlastic via GitHub [16] has been crucial for advancing our underwater waste detection efforts.

1.1.2 Object Tracking Integration

Ghaniazidani et al. [20] present pairing YOLOv8 and DeepSORT [21], which improves tracking reliability with the SORT algorithm, improving performance under challenging conditions like occlusions. BoxMOT offers tracking modules tailored for segmentation, object detection, and pose estimation, including seamless integration with YOLOv8[21]. These methods show YOLOv8-DeepSORT achieving higher tracking accuracy and fewer identity switches than YOLOv5 and SORT [20].

1.2 Methodology

Our pipeline extends marine debris detection and tracking algorithms and pairs them with our lab's existing underwater manipulation framework.

1.2.1 Detection

To build a reliable model for detecting plastic bags, we focused on gathering diverse visual data. Preliminary training data was collected in Seattle's Ship Canal. We varied the plastic bags in color, size, and quantity to capture a realistic distribution of bags that may be encountered in a real-world environment. Additional video was collected from our test tank and added to improve in-tank accuracy and diversify training conditions Figure 1.1).

This video was segmented into individual images and hand-annotated in RoboFlow [22]. Each segmented image was annotated with the class 'Plastic_Bag' by labeling the plastic bags visible in the images. This ensured that the bounding boxes accurately encompassed the plastic bags present in each image. The resulting dataset had 306 images. 74 annotated images were set aside for validation, and 44 were set aside for testing. The dataset was exported into the YOLOv8 format and named FindingTrash. To avoid confusion, it is important to note that this dataset is used to train the model we have named FindingPlastic.

Figure 1.1: Sample of images from the FindingTrash dataset

To incorporate our curated plastic bag dataset into existing models, we began by training YOLOv8 on the available DeepTrash dataset. We then used these trained weights as a starting point for our next training session with the FindingPlastic dataset. Using these weights allowed us to adapt the model for our specific detection task and accelerate the model's adjustment to detecting plastic bags.

We trained on the FindingTrash dataset, which was specified in a YAML file from Roboflow. The training configuration included 100 epochs with an image size of 416 pixels. The worker parameter was set to 2 for data loading to maximize processing time, and the batch size was adjusted to 16 to balance memory limitations and training pace. To promote faster convergence while preserving stability, the momentum was set to 0.9, and the starting learning rate was set to 0.001. To penalize excessive weights and prevent overfitting, a weight decay of 0.0005 was used. To lower the chances of divergence in the early phases of training, a warm-up period of three epochs was also added. The optimizer was set to 'auto', allowing the model to choose the best optimization strategy for the given training configuration. These parameters are presented in Table 1.1.

Table 1.1: Training Configuration for FindingTrash Dataset

Parameter	**Value**
Epochs	100
Image Size	416 pixels
Batch Size	16
Workers	2
Learning Rate (initial)	0.001
Momentum	0.9
Weight Decay	0.0005
Warm-up Epochs	3
Optimizer	auto

The evaluation metrics chosen were the mAP50 and F-1 scores. These were used in similar studies and provided a clear assessment of object detection performance. To understand these metrics, several terms must be introduced.

True Positives (TP) and True Negatives (TN): When the model accurately detects a positive class, it is said to be a true positive; conversely, when the model correctly detects a negative class, it is said to be a true negative.

Precision and Recall: These measures assess how well the model recognizes plastic in a picture. The definition of precision is:

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

Where TP represents true positives and FP denotes false positives. Recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where FN stands for false negatives.

Mean Average Precision (mAP): This metric assesses the model's capability to detect plastic across a set of images. It involves plotting a precision-recall (PR) curve and calculating the Intersection over Union (IoU) using the formula:

$$\text{IoU} = \frac{\text{BBox}_{\text{pred}} \cap \text{BBox}_{\text{gt}}}{\text{BBox}_{\text{pred}} \cup \text{BBox}_{\text{gt}}}$$

Where $\text{BBox}_{\text{pred}}$ and BBox_{gt} are the areas of the predicted and actual bounding boxes, respectively. Accurate detection requires setting high thresholds for both confidence and IoU. The mAP is then obtained by integrating the precision-recall curve:

$$\text{mAP} = \int_0^1 p(x)\,dx$$

Specifically, mAP50 calculates average precision at an IoU threshold of 0.5, providing insight into the model's performance with moderate overlap between predicted and ground truth boxes [15].

F1-Score: This metric provides a balanced measure by combining precision and recall into a single score, reflecting the model's overall performance [15].

1.2.2 Tracking

Following the development of object detection, tracking was needed to predict the motion of the target and to monitor multiple bags. The DeepSort tracking algorithm [21] applies the YOLOv8 FindingPlastic model to the segmented images. The algorithm assigns unique identification tags to the detected instances and applies Kalman filters for state estimation. The outputs are the four corners of the bounding box and the instance ID.

For the operation of the robotic arm, we utilized ROS1 (Robot Operating System) [10]. A series of nodes and packages were used to facilitate communication between various components of the system. Video from our test tank was captured and stored in a ROS bag file. Video data collected depth images, rectified images from the left and right cameras, and camera calibration information for the left camera. This file was played back to extract the necessary image sequences, simulating real-time data capture from our camera.

The tracking algorithm was applied to the collected tank data. The centroid of the bounding box was selected as the grasping point. For each image instance, the x and y pixel coordinates of the centroid were used to obtain depth information from the corresponding depth image. By combining the x and y pixel coordinates with the calculated depth, a 3D coordinate (x, y, z) was obtained for each image instance. For a bounding box defined by coordinates (x_1, y_1) and (x_2, y_2), the centroid (cx, cy) was computed as:

$$cx = \frac{x_2 - x_1}{2} + x_1 \tag{1.1}$$

$$cy = \frac{y_2 - y_1}{2} + y_1 \tag{1.2}$$

Equations 1.1 and 1.2 show the calculations of the midpoints for the x and y coordinates, respectively.

To integrate these coordinates into the robot's operational framework, a transformation process was necessary. The 2D-pixel coordinates needed to be transformed to align with the z coordinate, ensuring the accurate representation of the object's 3D position within the robot's coordinate system.

For each detected object instance, after calculating the centroid (cx, cy) of the bounding box defined by corners (x_1, y_1) and (x_2, y_2), the x and y pixel coordinates of the centroid were used to extract depth Z from the corresponding depth image. The camera intrinsic matrix $\mathbf{K}$ and the depth Z were then used to compute the precise 3D coordinates (X, Y, Z) of the object centroid. The camera intrinsic matrix $\mathbf{K}$ relates the pixel coordinates to the real-world coordinates as shown in Equation 1.3:

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1.3}$$

where f_x and f_y are the focal lengths along the x and y axes, c_x and c_y are the principal points, and s is the skew coefficient (if any). The 3D coordinates (X, Y, Z) of the object centroid are computed using Equation 1.4:

$$X = \frac{(cx - c_x) \cdot Z}{f_x}, \quad Y = \frac{(cy - c_y) \cdot Z}{f_y} \tag{1.4}$$

A point cloud of the collected tank data was produced for visualization. A script processed the ROS bag files to convert depth images into point clouds. It took in depth images, camera info, and RGB images. Using CvBridge, it converted ROS images to OpenCV format. The camera intrinsics were then scaled to match the depth image dimensions. The scaling was done using the following formulas:

$$f'_x = f_x \times \frac{w_d}{w_o} \tag{1.5}$$

$$f'_y = f_y \times \frac{h_d}{h_o} \tag{1.6}$$

$$c'_x = c_x \times \frac{w_d}{w_o} \tag{1.7}$$

$$c'_y = c_y \times \frac{h_d}{h_o} \tag{1.8}$$

Equations 1.5, 1.6, 1.7, and 1.8 show the adjusted focal lengths and principal points based on the new dimensions.

Where f_x and f_y are the original focal lengths, c_x and c_y are the original principal points, w_o and h_o are the original width and height, and w_d and h_d are the depth image width and height. These scaled parameters were used to transform depth images into 3D point clouds with PyTorch.

RGB images were converted to grayscale to provide intensity values for the point clouds. The final point clouds, containing 3D coordinates and intensity data, were formatted into PointCloud2 messages and written to the output ROS bag file.

Additionally, a separate script was developed to continuously publish the centroid using ROS markers.

This script was synchronized with the point cloud frame to ensure detailed visualization.

1.2.3 Object Acquisition and Storage

To enable real-time grasping, integration between the arm's motion and the detected plastic bag was achieved using the Servo() class from MoveIt Servo [23]. The tracked centroids were converted into Pose Stamped messages and published as target positions. The arm subscribed to these messages and continuously planned paths to reach the most recent centroid location of the bag in real time.

During the acquisition phase, the Bravo arm [24] will retrieve plastic bags using a specialized gripper. In our case, the gripper will be a suction gripper. This choice should reduce the chances of slippage or misalignment. A sensor system will confirm the collection and storage of the bags.

In the storage phase, after securing the bag, the Bravo arm will carefully store it for later retrieval during surfacing operations.

1.3 Results and Discussion

For object detection, we wanted to train a model using YOLOv8 architecture on the existing DeepTrash dataset. Despite efforts, replicating the YOLOv5 DeepPlastic model's high performance with the newer YOLOv8 trained on the same dataset proved challenging. Regardless of having access to more hyperparameters, we were not able to replicate the high mAP50 reported using YOLOv5. The best performance we recorded using YOLOv8 trained on DeepTrash was 76% mAP50. This was not nearly as high as the established DeepPlastic weights. Subsequently, we reverted to YOLOv5 and utilized the DeepTrash provided best weights [14], replicating the previously observed 91% mAP50. Our evaluation involved testing the dataset utilizing the built-in YOLO validation feature.

As object detection evolved, continuous detection and movement prediction of plastic bags became necessary, requiring tracking. Initially, YOLOv5 was considered for tracking. However, integrating newer tracking methods with the older YOLOv5 model presented compatibility challenges, prompting a shift to YOLOv8.

For improving the YOLOv5 DeepPlastic model, we utilized the pre-trained weights from DeepPlastic, which had an impressive performance of 91% mAP50. These weights were employed as the initial starting point for the training of our new model using the FindingTrash dataset. This method resulted in an improved mAP50 of 96%. However, the same method using YOLOv8 architecture on the FindingTrash dataset showed an improved 98% mAP50 and 98% F-1 Score. These metrics were promising, and the visual output is shown in Figure 1.2. Thus, we opted to continue with our FindingPlastic model trained with YOLOv8.

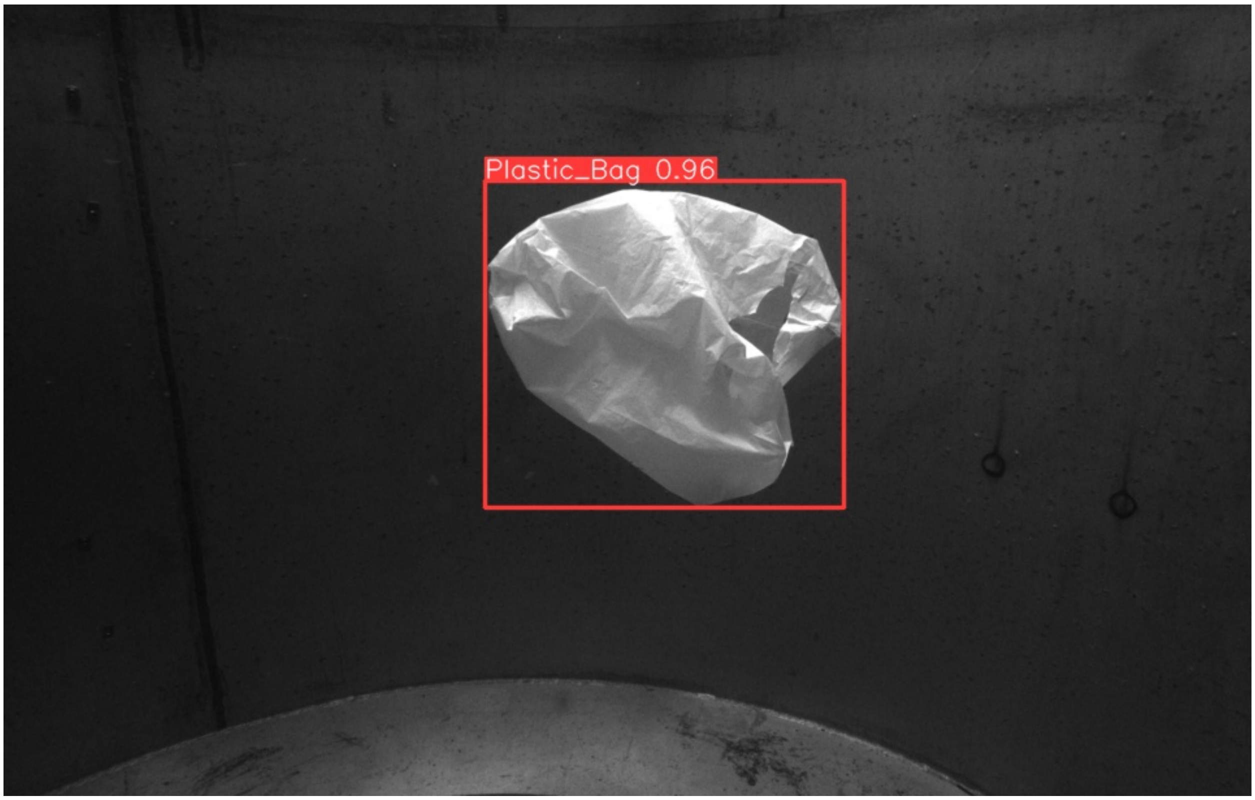


Figure 1.2: Visual output from YOLOv8 detection on segmented images from recorded test tank data.

The following table provides a comparison between YOLOv8 and YOLOv5 when trained on each of the DeepTrash and FindingTrash datasets.

Table 1.2: Performance Comparison of YOLO Models on DeepTrash and FindingTrash Datasets

Dataset	DeepTrash		FindingTrash	
	mAP50	F1 Score	mAP50	F1 Score
YOLOv8	76%	0.74	98%	0.98
YOLOv5	91%	0.89	96%	0.96

As shown in Table 1.2, the YOLOv8 FindingPlastic model demonstrates the best performance, supporting the decision to upgrade to YOLOv8. Overall, the FindingTrash dataset displays higher F1 scores and mAP50 across both YOLO models. This shows a clear improvement from existing models to the FindingPlastic model. It should be noted that the FindingTrash dataset contained fewer images during training than that of DeepTrash, which may have contributed to the model's observed higher performance. The lack of image diversity may have simplified the detection tasks, making it easier for the model to achieve higher results. Moreover, the model was trained with only a single item class, which could have further enhanced its performance by focusing on a singular specific detection goal.

Testing FindingPlastic with multiple, varying bags yielded similar results to singular bag outputs. However, smaller, clear bags blended more seamlessly with their surroundings. This can be seen in Figure 1.3, where the confidence score for the smaller clear plastic bag can be seen at a much smaller 54% confidence when compared to surrounding detections. Another issue that was addressed was that the replicated DeepPlastic model would struggle with reflections. This was solved by incorporating more images that featured reflections during training, featured in the FindingTrash database. Using a higher confidence interval could help mitigate these issues by considering only highly confident detections, and filtering out false positives.

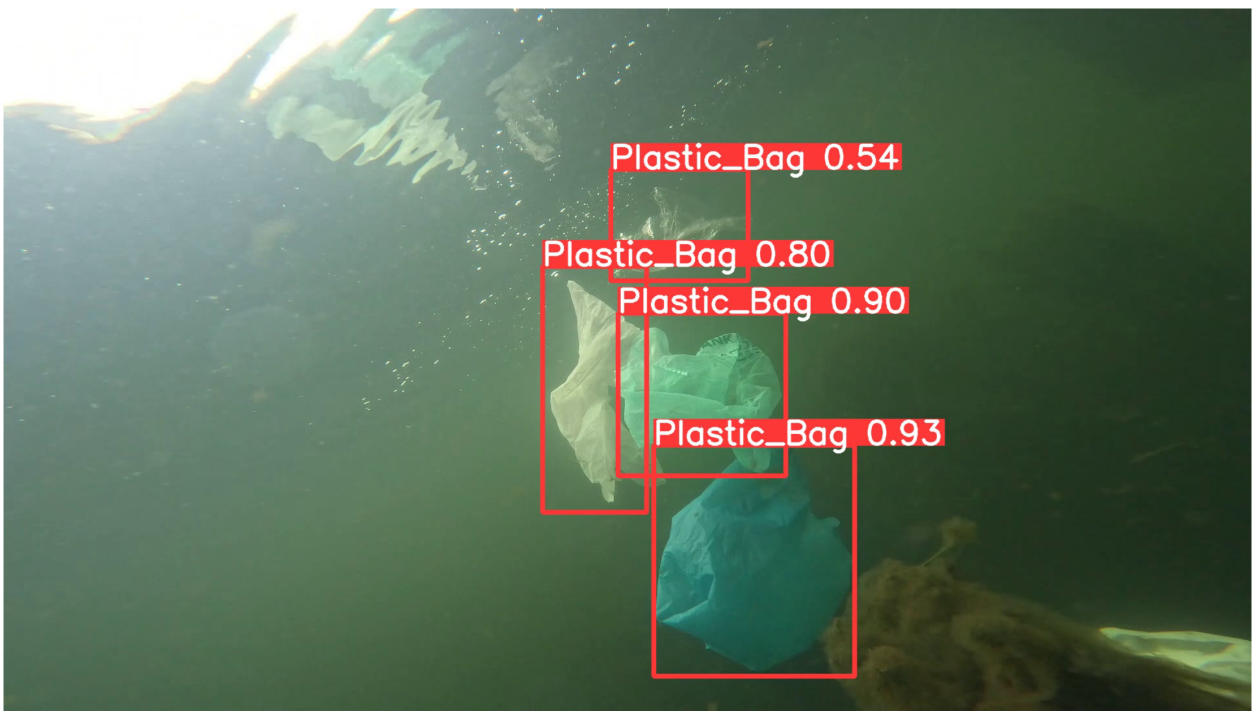


Figure 1.3: YOLOv8 detection output of four plastic bags, including a smaller, clearer bag at 54% mAP50.

Additionally, the visual output of the DeepSORT tracking algorithm can be seen in Figure 1.4. This visual displays multiple tracking identifications in this case, 1-3, and confidence scores above 90 % in turbid waters and on a white, blue, and green plastic bag.

Figure 1.4: Output from DeepSORT tracking showcasing three individual identification tags.

Following the successful demonstration of the tracking algorithm on the captured data, methods to integrate this tracking with the robotic arm were explored.

1.3.1 Qualitative System Evaluation

To assess our integrated system's effectiveness, trials were simulated virtually in our test tank environment. This consisted of publishing a marker of the x, y, and z coordinates representing the centroid of the bag to a simulated test tank and observing if these points overlapped our recorded bag as it drifted through the tank. In this simulation, the observed centroid points were published as markers along with a bag file to provide visual confirmation that the markers followed the same path as the video of the plastic bag. A still image captured from this process can be seen in Figure 1.5.

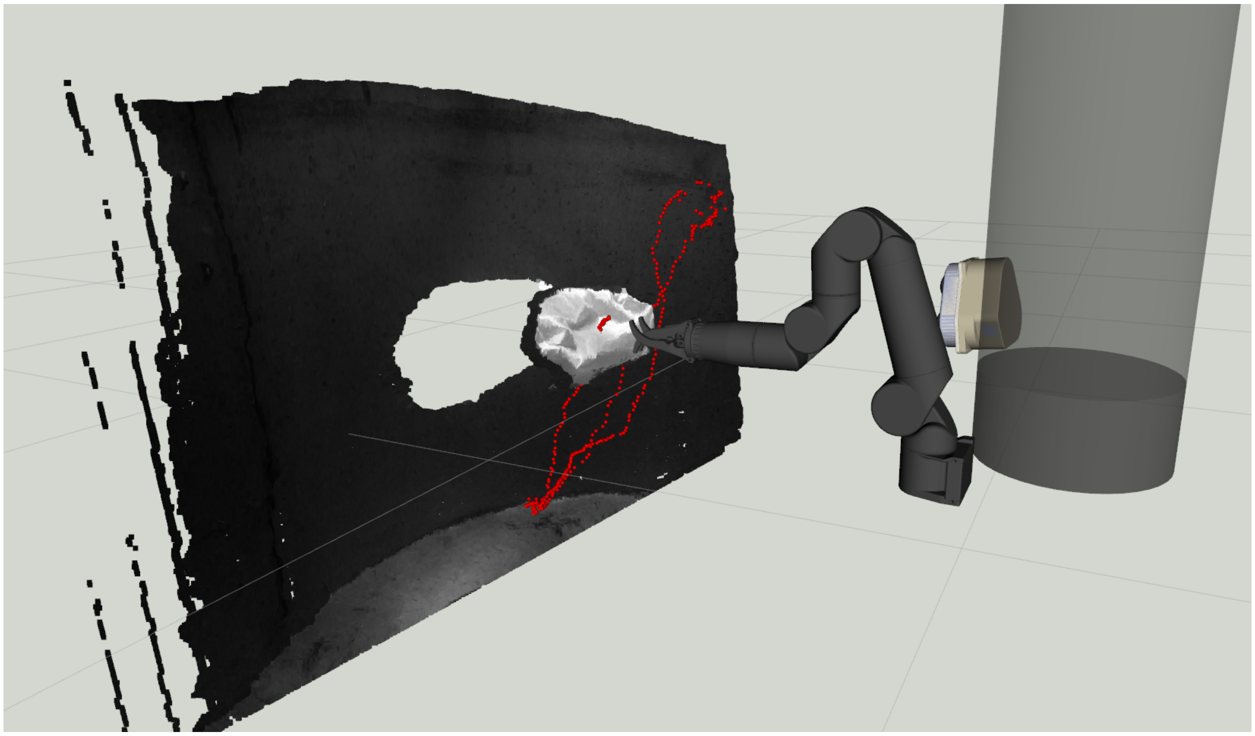
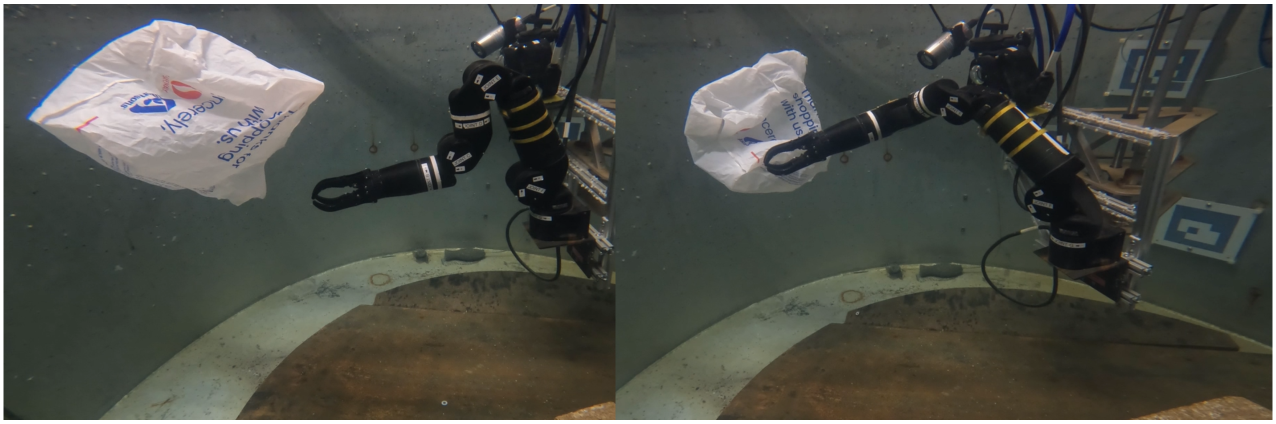


Figure 1.5: Simulation of test tank with point cloud generation of recorded bag video with calculated centroid position (in red) overlayed.

The bag was successfully tracked, as indicated by the trail of red markers, which closely matched the plastic bag's location in the point cloud. This allowed us to confidently use these marker points to guide our manipulator.

To evaluate the system's real-world capabilities, we ran a series of trials in a controlled test tank environment. These tests focused specifically on the system's ability to track and follow drifting plastic bags using real-time centroid updates within our servoing framework. Our main objective at this stage was to evaluate tracking-driven motion. The planned specialized suction gripper has not yet been integrated. This gripper is expected to improve grasp success by increasing the effective capture area through suction. As a result, grasping and storage were not emphasized, and completed grasps were not expected. Instead, the focus was on assessing whether the system could detect, track, and reach a moving target.

Images from the tank evaluation, shown in Figure 1.6, highlight the system's ability to reach the bag from its stowed location to the target. The manipulator followed the bag's path and made consistent contact with the bag. This outcome validates the core behavior we aimed to test. These results demonstrate that the autonomous tracking and motion control operate effectively in real time and under real-world conditions.



(a) Robotic arm in ready position. **(b)** Qualitative grasp completed: Contact made.

Figure 1.6: Qualitative grasp demonstration in the test tank. The left image shows the arm in its initial ready position, while the right image shows the arm extended to the bag's location, making contact.

1.4 Summary

In this chapter, we track, detect, and remove plastic bags from marine environments using robotic arms and computer vision. Visual tracking combined with integrating YOLOv8 improves monitoring and makes bag recovery and removal more effective. Pairing this with our Bravo underwater arm will offer a practical solution for removing plastic bags from dynamic underwater environments. Tests conducted in our simulated and real-world environments confirm that our system effectively detects and tracks plastic bags, demonstrating its potential capability for removal in real-world situations.

Chapter 2

Autonomous Capture of Dynamic Plastic Bags Underwater

Chapter 1 introduced vision-based methods for detecting and guiding a subsea manipulator to reach for plastic bags. This system is purely reactive and can only reach the bag's current position. To handle the challenges of dynamic bag motion, the algorithm was updated to predict future bag locations by estimating bag velocity and trajectory. This also allows the determination of whether a bag is reachable in the time available before initiating motion. We now integrate that detection and tracking foundation to handle drifting plastic bags, as they would appear in the ocean. To simplify the problem, we assume only vertical motion. This is a reasonable assumption, as plastic bags tend to be nearly neutrally buoyant [25] and are primarily affected by gravity and slight currents. In this chapter, our proposed system addresses how optical sensing can inform motion planning in autonomous robots to track and intercept drifting plastic bags.

The method described here has three core components: motion prediction, informed grasping, and grasp planning that integrates both to enable timely interception.

2.1 Related Work

Manipulating dynamic objects underwater remains a complex challenge due to uncertain object and platform motion, sensing limitations, and the difficulty of precisely positioning an end-effector in real time. Unstructured environments, 6-DOF object movement, and unpredictable currents complicate motion prediction. These issues are further exacerbated by visual degradation from poor lighting and turbidity, and the lack of fixed environmental references limits the reliability of traditional tracking and prediction methods[26].

Vision-based sensing offers a practical and informative solution, particularly when paired with stereo cameras that provide depth information essential for estimating object position and motion [27]. Object detection models such as YOLO have shown strong performance in marine debris identification, offering the real-time inference needed for onboard robotic systems [18]. Prior work has incorporated specialized datasets of plastic bags into autonomous underwater retrieval pipelines using manipulators [28], highlighting the value of integrating visual systems with robotic manipulation.

Recent research has also explored the integration of visual systems into robotic manipulation of dynamic targets. Marturi et al. [29] proposed techniques for updating grasp plans in response to object motion, combining predictive and reactive strategies. Similarly, Huang et al.[30] introduced learning-based methods that adapt to uncertain object dynamics during manipulation, reinforcing the value of real-time perception in closed-loop control.

However, most of these advances focus on interacting with dynamic objects on land or in the air. The challenge of manipulating deformable, drifting objects, such as plastic bags, underwater remains largely unaddressed. Inspired by previous visual motion prediction systems, we designed a simplified dynamics predictor for underwater conditions. Our system pairs real-time visual tracking with a reachability-based planner to anticipate object drift and inform manipulation. This approach bridges the gap between detection and action, offering a practical solution for dynamic underwater tasks.

2.2 Methods

This work builds on the FindingPlastic detection model to track and identify plastic bags [28]. Using YOLO and DeepSORT, this model was integrated into our lab's existing framework for underwater manipulation. The system includes a 7-degree-of-freedom manipulator and a mounted integrated stereo camera. The full system setup is shown in Figure 2.1.

This chapter extends on the previous system by focusing on predicting future grasp locations of drifting plastic bags. We specifically address a simplified vertical motion within the water column, assuming that

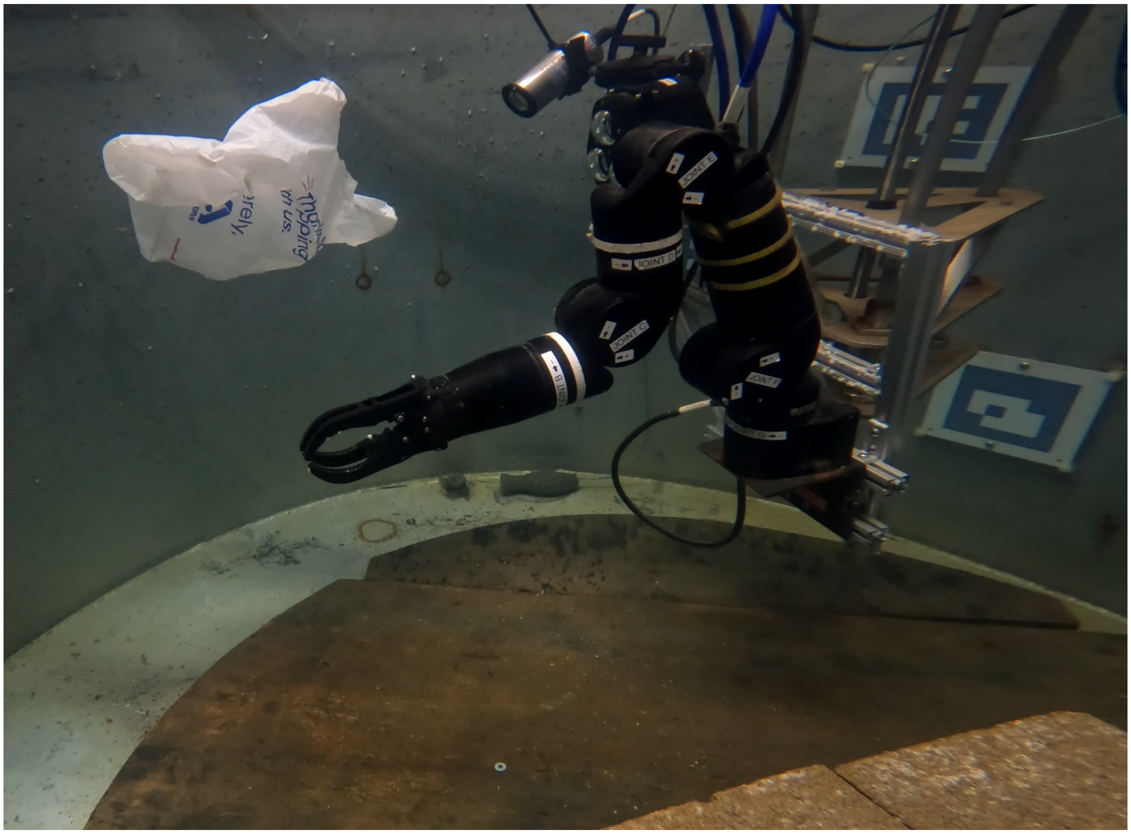


Figure 2.1: Existing underwater manipulation testbed with Bravo Reach 7 arm and stereo camera.

the plastic bag moves only along the vertical (Z) axis. This simplification enables the development of a basic trajectory planner, which can be expanded later to handle more complex dynamics. Unlike earlier approaches that continuously follow the object, our system anticipates the bag's descent and predicts a future interception point. Then, the manipulator can drive to the interception point ahead or simultaneously with the target bag.

Real-time manipulator control was handled using the Mover class from MoveIt Servo, which converts target poses into joint velocity commands for smooth, continuous motion. Built on the ROS 1 framework, this setup allowed the arm to communicate and execute predicted grasp positions in real-time.

2.2.1 Motion Prediction for Vertical Drift

For identifying plastic bags, we integrated our FindingPlastic model to detect and track plastic bags within our camera frame. From the detections, we assume the centroid to be a graspable position, which results in a 3D point in our robot's coordinate frame. We calculate the 3D centroid of each tracked object by unprojecting the 2D bounding box center using the depth image. For motion prediction, a standard kinematic model was used to predict future positions of the bag based on vertical speed and the initial detected position of the bag's centroid. The kinematic model can be seen in (2.1) where X_f is the calculated vertical position:

$$X_f = X_0 - v_0 t \tag{2.1}$$

2.2.2 Informed Grasping: Reachability Dataset Construction

To inform the robot's decision-making for grasping, a reachability dataset was constructed for the robotic arm. Each entry includes 3D coordinates of a potential end-effector pose, a flag indicating reachability, and the execution time if reachable.

The dataset was created by systematically sampling the robot's workspace in simulation. Simulation of the robotic workspace was performed within ROS using RViz, which provided a real-time visualization of the robot's state and environment. The simulated test tank is shown in Figure 2.2.

Since this work was conducted in simulation, motion was not executed to reduce runtime. Rather, for each position, we used the estimated plan times from Moveit Servo. For each sample, one degree of freedom was varied while the other two were swept through fixed values. The orientation of the end effector was set in a neutral pose, and the tolerance for the orientation of the goal was relaxed to 3.14 radians (180°). This increased flexibility allowed the planner to fully access the robot's reachable workspace, as it was not constrained to a fixed orientation.

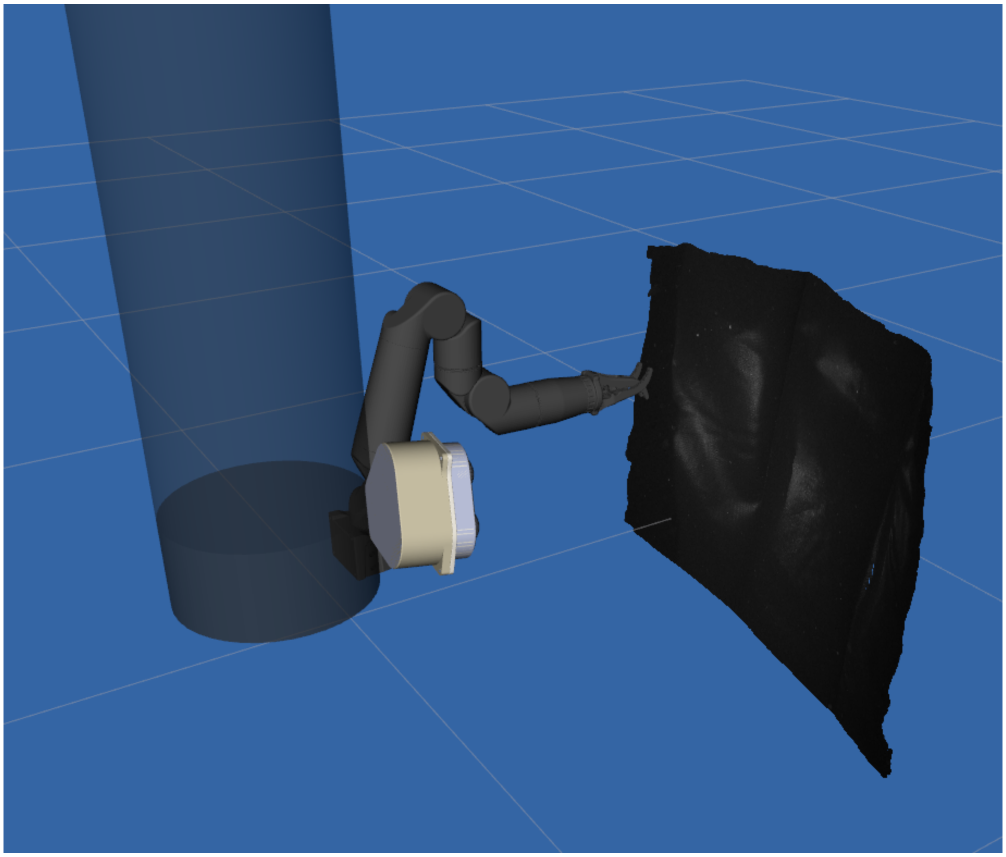


Figure 2.2: Simulated set-up of test tank in RViz, featuring point cloud representation of recorded data from test tank.

Based on the arm's physical limits, we selected a region that slightly exceeded its typical reach to ensure the full workspace was captured. Planned motions were executed across a Cartesian grid bounded by: $X \in [0.0, 0.8]$ m, $Y \in [-0.8, 0.7]$ m, and $Z \in [-0.4, 1.0]$ m, sampled at 0.1 m intervals. For each planned point, motion was attempted. We recorded whether the goal was successfully reached and the duration of the motion. Sparse regions in the resulting dataset were interpolated using a NumPy-based 3D method, assuming that any positions inside the observed edges were reachable.

2.2.3 Grasp Planning: Interception matching

The initial step toward integrating predicted trajectories and reachability involves the planner constraining the x and y coordinates of the first detected centroid. The z coordinate is sampled downward through the reachable vertical range to form candidate grasp positions.

The vertical velocity v_0 is computed by taking the difference in the object's height between frames and dividing by the elapsed time. We maintain a rolling buffer of recent velocity estimates for each track to smooth out noise and use the averaged result from the time of detection.

We first predict the object's future positions along the vertical axis by sampling a range of z values below its current height. For each sampled z, we estimate the time it will take the object to reach that position using the vertical motion model from Equation (2.1). This results in a set of $(z, \text{arrival time})$ pairs that describe when the object is expected to pass through each depth.

Next, we use these same z values to query the precomputed reachability dataset, which provides the time required for the robot to reach each corresponding depth from its current configuration. This yields a matching set of $(z, \text{time-to-reach})$ pairs from the perspective of the arm.

By comparing the arrival time of the object at each z with the robot's time-to-reach, we identify viable interception points. A point is considered valid if the robot can reach it before the object arrives and if it lies below the object's current height. Once the first feasible interception point is identified, the system proceeds to execute the grasp using the robotic manipulator.

2.2.4 Acquisition and Storage

We plan to use the Bravo arm equipped with a suction gripper to retrieve the plastic bags, minimizing the risk of dropping or improper grasping. A sensor system will be used to confirm successful collection. Once secured, the bag will be stored by the arm for later retrieval during surfacing operations.

2.3 Experiments

2.3.1 Intercept Prediction via Simplified Dynamics

To characterize the system's ability to intercept moving targets, we implemented a simplified one-degree-of-freedom model. In this formulation, we fixed the object's horizontal position (x_0, y_0) and assumed no lateral drift, setting $v_x = v_y = 0$. The detector provided the initial position at the time of first detection, and vertical velocity was estimated using our tracking module.

The first experiments considered for the testing of the effectiveness of the system were done in simulation. Multiple ROS1 bag files were recorded to capture instances of plastic bags descending through the test tank.

Experiments were conducted at five positions along the x-axis, capturing the bag from varying viewpoints, ranging from highly occluded to fully visible and back to partial views on the opposite side. Each instance began with nothing in frame. A singular plastic bag was placed within our test tank with weights placed within the bag. These weights allowed the bag to descend through the visible portion of the test tank, mostly along the z-axis. This weight was chosen to overcome the naturally buoyant tendencies of the bag. The path of the bag was carefully selected to start with the bag out of frame and have it drift downwards into frame and then out again. Having these recordings allowed the system to test on controlled data repeatably. By playing the rosbag back in rviz we were able to have a simulated test tank environment to test the capabilities of the system repeatably. The set-up of this experiment can be seen in Figure 2.3

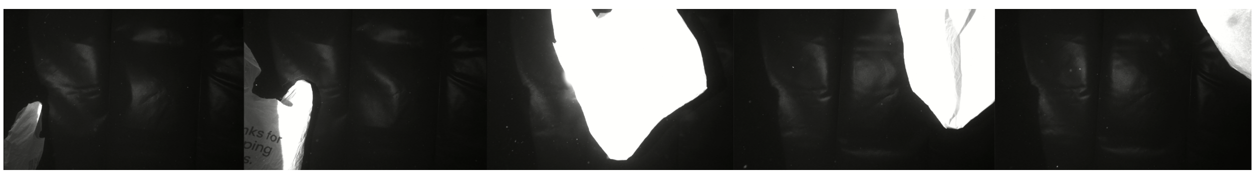


Figure 2.3: Varying five positions along horizontal axis for testing. Position 5 through Position 1, shown left to right.

2.3.2 Evaluation Metrics

To evaluate the system's performance, we defined a metric for successful grasp completion. For our task, a grasp is considered successful if it occurs within half the bag's vertical length from its centroid. Given that the plastic bags are 53 cm long, this translates to a ±26.5 cm tolerance. To assess this, we compared the predicted and observed positions of the object at the moment the grasp was attempted. We recorded the timestamp when the end-effector reached its goal pose and compared it to the tracked centroid position from the continuously running tracker script at that same time.

Since the system currently predicts motion only along the z-axis, we restricted evaluation to a single degree of freedom. This focused metric provides a straightforward, quantifiable measure of the system's prediction accuracy along the vertical axis.

2.4 Results

The reachability dataset was visualized as a 3D heatmap, as shown in Figure 2.4, showing time-to-goal from different positions in the workspace. As expected, areas near the edges of the robot's reachability envelope took longer, and areas outside the hull failed entirely. In contrast, all positions within the hull were consistently reachable.

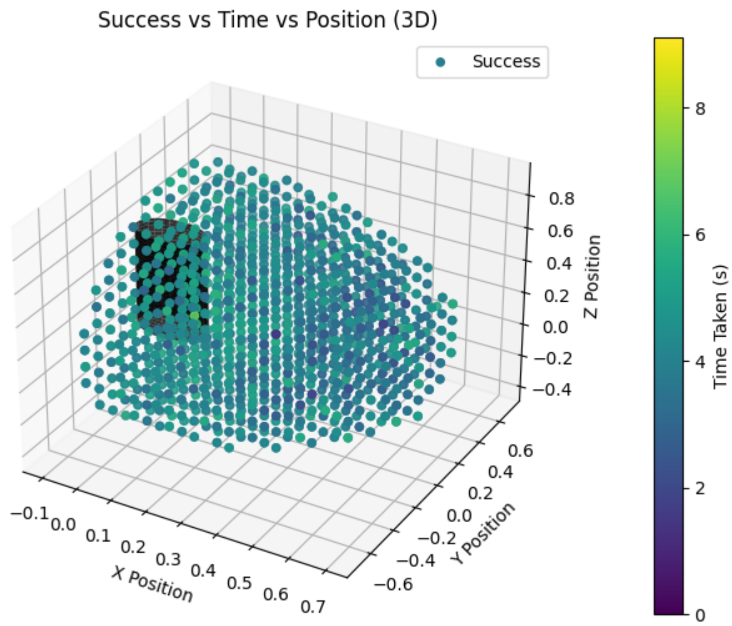


Figure 2.4: Plotted heatmap of reachable end-goals with time association.

2.4.1 Grasp Logic: Comparison of Z-values

After receiving the initial centroid position, the system fixed (x_0, y_0, z_0) as the starting point for generating candidate grasp positions. Among these, the system selected the first position along the bag's descent trajectory where the robot's estimated time-to-reach was less than the predicted time of arrival of the bag. This interception occurred at the earliest feasible opportunity.

An example of this decision-making process is shown in Table 2.1. The highlighted point indicates the first z-position where both temporal and spatial constraints were satisfied. After populating an array with time values from 0 to 10 seconds, the corresponding z-positions of the drifting bag were estimated at each step. The velocity used was assumed to be 0.01 m/s for this example. These z-values were then used to query the precomputed time-to-reach dataset. Although this condition was met at later positions as well, the first point where the interpolated time-to-reach fell below the predicted arrival time was at $z = 0.227\,\mathrm{m}$. As a result, the system selected this z-coordinate, along with the fixed x_0 and y_0, as the grasp point.

Table 2.1: Comparison of predicted bag arrival times and interpolated robot reach times at each vertical position z. The selected grasp point is highlighted.

Predicted Bag Arrival Time		**Interpolated Robot Reach Time**	
z [m]	Time [s]	z [m]	Time [s]
0.267	0	0.267	3.049
0.257	1	0.257	2.934
0.247	2	0.247	3.083
0.237	3	0.237	3.026
0.227	**4**	**0.227**	**3.015**
0.217	5	0.217	3.046
0.207	6	0.207	2.995
0.197	7	0.197	2.926
0.187	8	0.187	3.044
0.177	9	0.177	2.986
0.167	10	0.167	3.101

Grasp selected at $z = 0.227$, where robot time-to-reach (3.015 s) is less than predicted bag arrival time.

2.4.2 Simulated Trials

Using the test tank simulation and recorded rosbags, we evaluated the prediction accuracy of the system across multiple trials, each with different initial drop locations and speeds. These variations tested the system under a range of dynamic conditions.

Five positions were evaluated for feasibility. At each position, three trials were run. During testing, it was found that Positions 1 and 2 resulted in end-effector goals that fell outside the convex hull of the dataset used for time interpolation. As a result, the system did not have data to estimate the time-to-goal for these points. This led to expected failures and a subsequent system shutdown. The remaining three positions were located within the reachability dataset's bounds and were used for further evaluation.

To evaluate performance, we plotted the vertical position z reached by the end-effector with the observed

z-position of the bag at the corresponding time. The results of this comparison are shown in Figure 2.5.

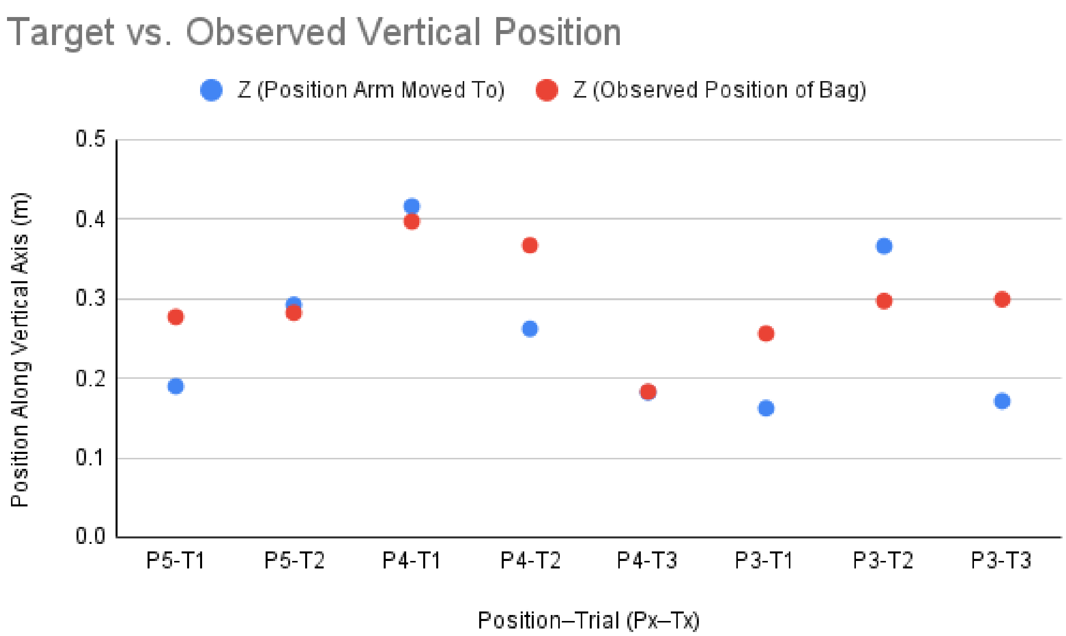


Figure 2.5: Comparison of target and observed vertical positions for all trials at Positions 5 to 3. Each position includes three trials, except for Position 5, where one trial was excluded due to a misdetection.

Since all trials exhibited vertical deviations within the ±26.5 cm tolerance defined for a successful grasp, the system would have grasped the plastic bag in all eight trials.

However, in 3 of these trials, the predicted positions exhibited unanticipated behavior. Based on the grasp algorithm, the end-effector was anticipated to reach a position at or slightly ahead of the bag's centroid at the same point in time. At Position 3, Trial 2, the bag was observed 6.9 cm below the attempted grasp point; at Position 4, Trial 1, it was 1.9 cm above; and at Position 5, Trial 2, it was 1 cm above.

We observed that the arm arrived too late in cases where the plastic bag's initial velocity was computed to be very low. This behavior was due to the velocity estimate being based only on the first few frames, which did not account for the bag's later acceleration. As a result, the system predicted a shallower descent and moved to a higher position, falling behind the bag's actual trajectory.

This trend is evident at position 3, where the only case of a late arrival is shown in red as seen in Figure 2.6. The mean velocity over time reveals that Trial 2 began with a much lower average velocity of approximately 0.03 m/s, while the other two trials maintained averages closer to 0.075 m/s. While all trials exhibited some velocity inconsistencies, only Trial 2 significantly underestimated the bag's motion early on. As the data stabilized, a higher and more consistent trend emerged, highlighting the initial underestimation.

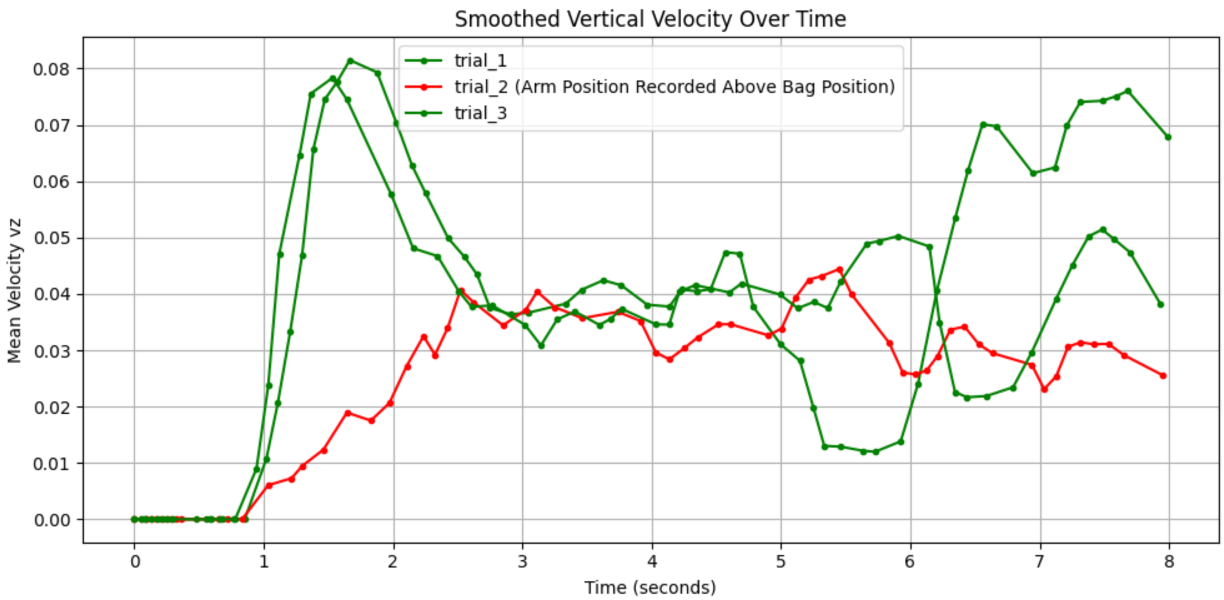


Figure 2.6: Velocity over time for all trials at Position 3. The outlier, highlighted in red, corresponds to the case where the arm moved above the bag's position, showing a noticeable divergence from the other trials.

Outlier cases at positions 4 and 5 showed discrepancies of 1.9 cm and 1.0 cm, respectively. In both cases, the end-effector reached a goal position above the bag's recorded position. These cases follow a similar pattern to what was observed at position 3. However, they appear more closely aligned with their corresponding trials. At position 4 (Figure 2.7), the first five detection instances in trial 2 show relatively low velocities compared to the later portion of the trial. This supports the earlier observation that grasps tend to fall short of bag positions when motion estimates rely on slower initial velocities. Trial 3 at the same position shows a similar pattern but with a shorter duration of low velocities, resulting in a grasp that was slightly ahead of the bag's position. The velocity patterns in these trials are quite similar, which makes the trend more difficult to discern. However, the underlying relationship likely still holds.

At position 5 (Figure 2.8), several low-confidence detections, likely due to partial views of the tarp, delayed the system's ability to detect the plastic bag consistently. Once confident detections resumed, the same velocity trend was observed in trial 2. The system used early low-velocity values for motion prediction, but the actual velocity increased throughout the trial. This caused the bag to drift farther than predicted, and it was ultimately observed at a lower position than the planned intercept, contributing to the arm arriving after the plastic bag.

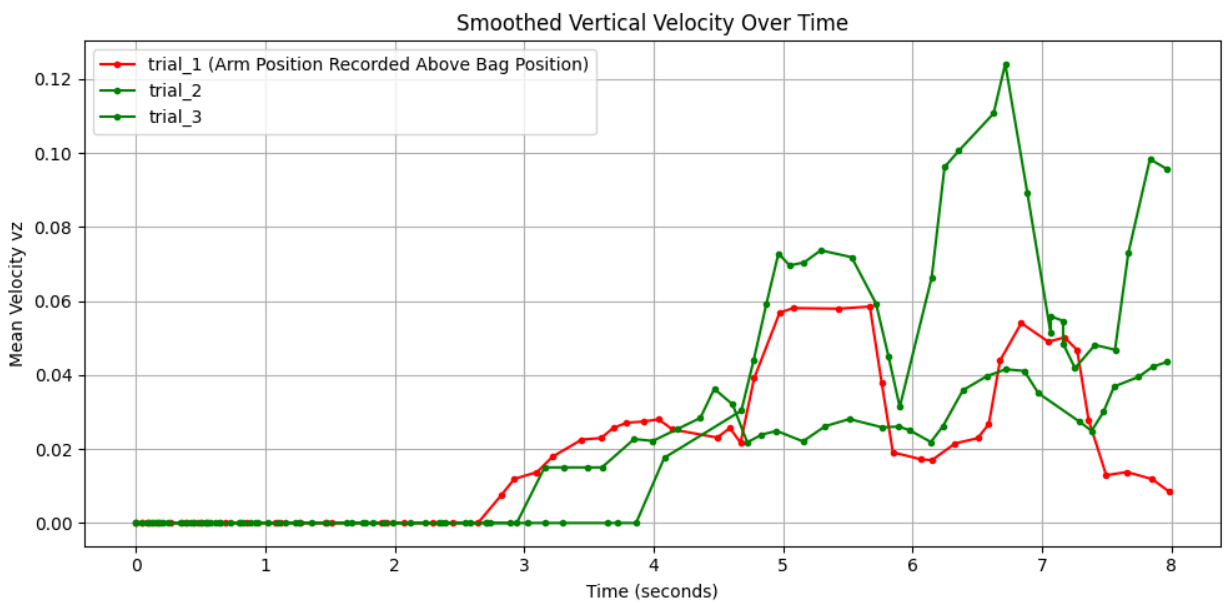


Figure 2.7: Velocity over time for all trials at Position 4. The outlier, highlighted in red, corresponds to the case where the arm moved above the bag's position, showing a divergence from the other trials.

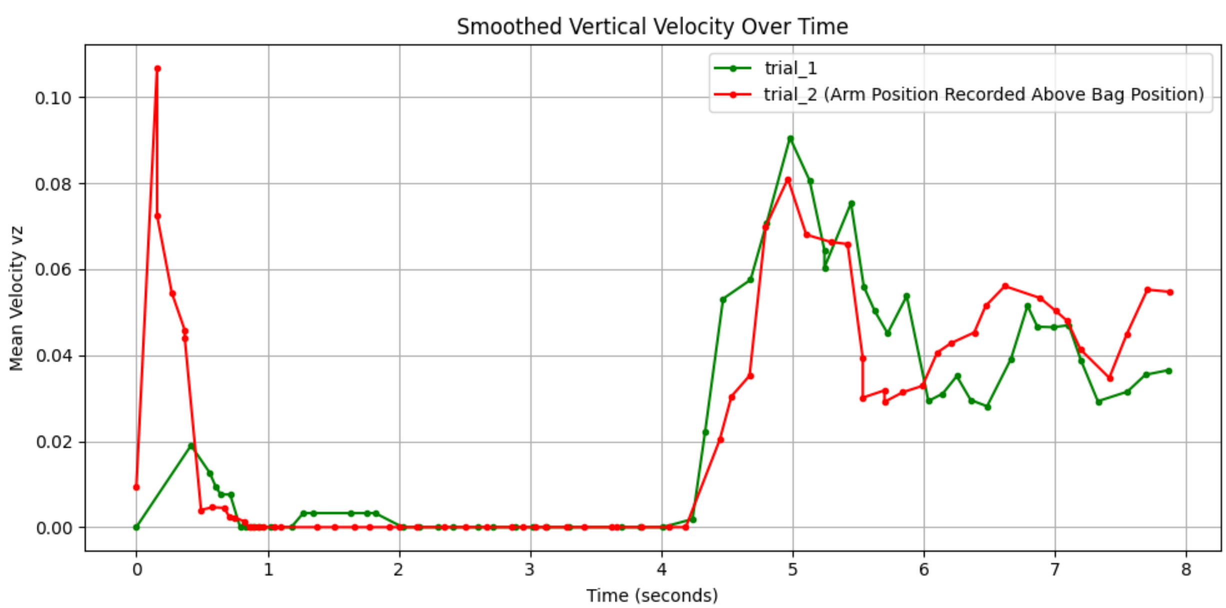


Figure 2.8: Velocity over time for all trials at Position 5. The outlier, highlighted in red, corresponds to the case where the arm moved above the bag's position, showing a divergence from the other trials.

Additionally, one failure was observed due to a misdetection. This occurred near the edge of the camera frame at position 5. The detector mistook rapid water movement combined with glare and texture from the tarp's backside for a plastic bag. The false detection appeared in one frame above the confidence threshold of 0.90, which caused the system to fail. This failure, seen in Figure 2.9, was a result of the detector from Chapter 1 and therefore not included in the data analysis.

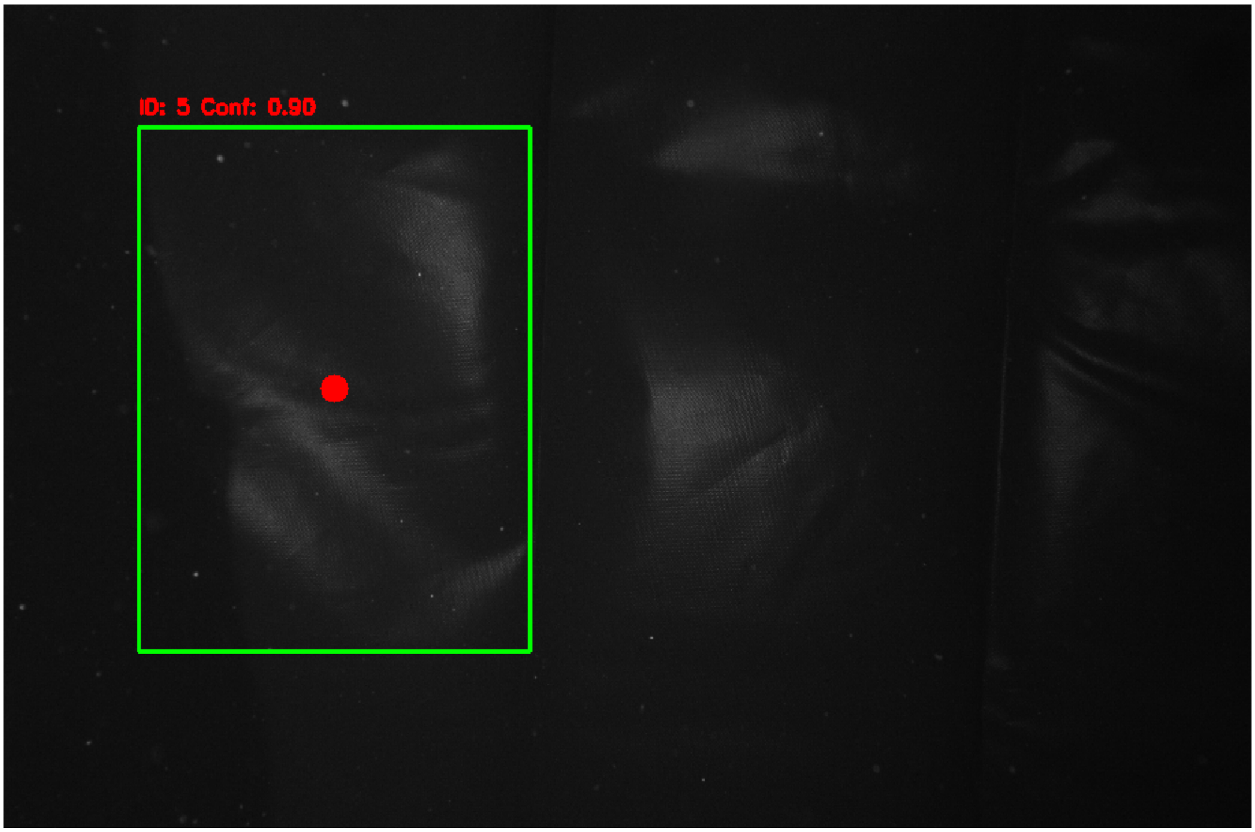


Figure 2.9: Detection failure: false positive due to glare and water motion near tarp edge.

2.4.3 Qualitative System Performance

This section presents a qualitative analysis of the most pronounced miss, observed at Position 3, Trial 2. The end-effector was recorded to be 6.9 cm above the position of the observed bag. Figure 2.10 shows that despite arriving 6.9 cm above the centroid of the bag, the arm still makes contact with the bag.

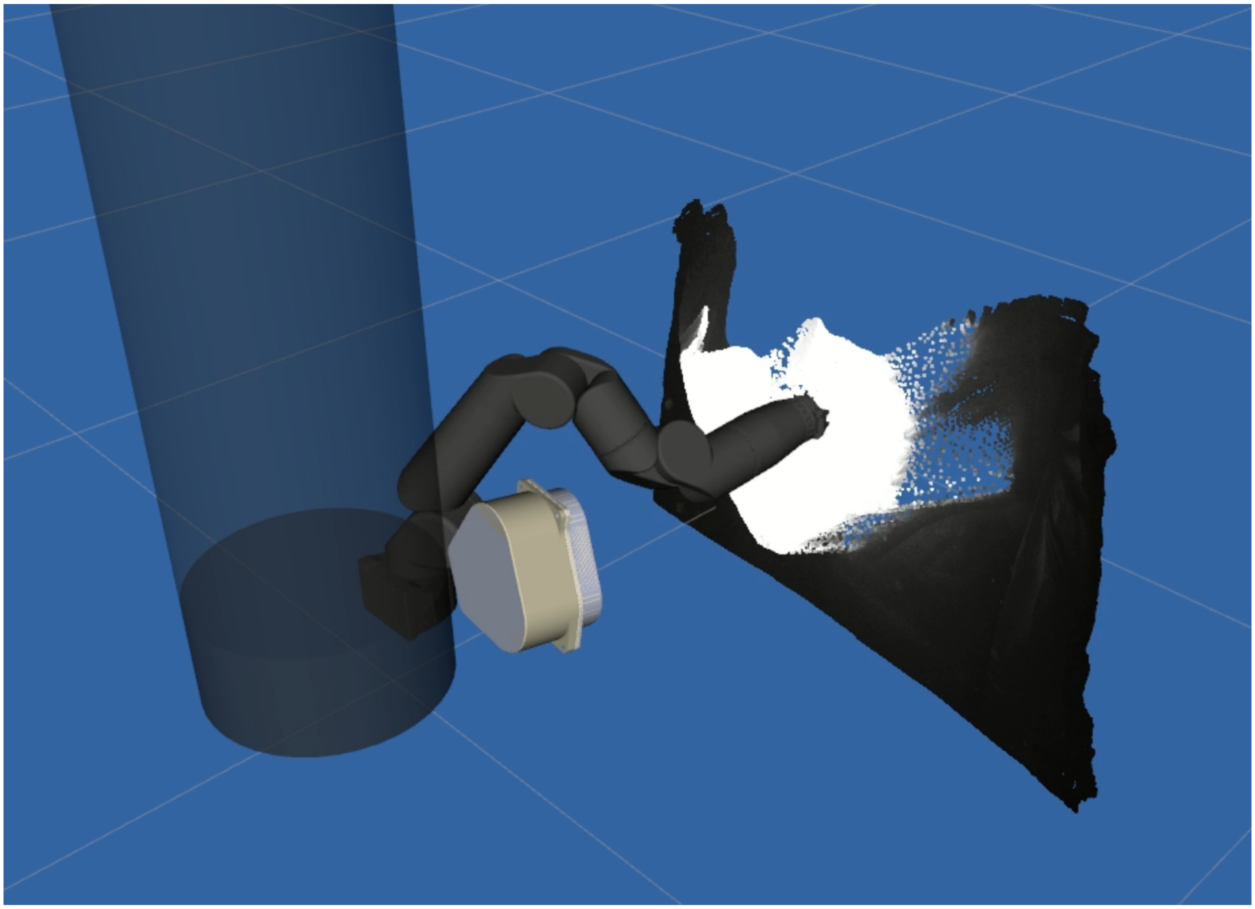


Figure 2.10: Qualitative analysis of grasp completion during simulation at Position 3, Trial 2.

2.5 Discussion

2.5.1 Causes of Grasp Prediction Deviations

Across multiple positions, trials with significant deviation consistently began with low initial velocity estimates. In these cases, vertical errors ranged from 1.0 to 6.9 cm. The velocity estimation module relied heavily on the first few frames and, when early motion was minimal, predicted a slow descent, failing to anticipate later acceleration. As a result, the end-effector arrived too high, missing the actual bag position.

Three of the eight trials showed this behavior clearly, with the arm arriving above and after the plastic bag had already passed through the predicted grasp point. These discrepancies point to a weakness in the motion prediction system that could limit performance in more precise tasks.

Together, these findings reveal a structural limitation in the system: early velocity estimates are overly influential and not sufficiently corrected as motion evolves. This occasionally leads to an underestimation of vertical motion when the object accelerates mid-trajectory. Such outliers may stem from the bag initially resting on the water's surface, where surface tension must be overcome before transitioning into free fall.

However, at Positions 4 and 5, the trials where the arm moved above the bag did not show velocity profiles that differed significantly from those in their respective tests. This suggests that early velocity estimates alone may not fully explain the missed grasp. In these borderline cases, partial occlusion near the edge of the field of view may have further contributed to the unexpected behavior.

2.5.2 System Performance

These results suggest that while the system performs reliably overall, some failures still occur. Of nine trials, one resulted in a clear failure, and three displayed unanticipated behavior. However, two of those were within 2 cm of the target. The largest positional error along the z-axis was 6.9 cm behind the bag. In Figure 2.10, this miss is contextualized within the task of completing a grasp of a plastic bag. This shows that while such errors are significant on the scale of individual centroids, they are small relative to the plastic bag's dimensions (53.34 cm × 29.21 cm). Within this context, the misses appear acceptable, and successful grasps would likely still occur.

2.5.3 Assumptions

Although our model does not fully capture the complexity of oceanic drift in real-world conditions, it provides a tractable starting point for evaluating real-time interception under dynamic constraints. The current kinematic motion model of the plastic bag can be further refined to accommodate more complex drift behaviors.

We used 3D interpolation to estimate the time required to reach target positions across the robot's workspace. This approach enabled us to generate grasp durations for positions between sampled points in the reachability map. Although this method improved spatial coverage, it also introduced some uncertainty.

To maximize reachability, the end-effector was given a neutral orientation with a relaxed 180° tolerance, allowing the planner greater flexibility to reach more positions. However, the loosened orientation constraint may result in varying approach angles for repeated movements to the same target.

The eight trials used to evaluate system performance were determined to be successful, under the assumption that full plastic bags were present. While we acknowledge that a bag may be compressed, crumpled, or partially intact, our collected and observed data did not include such cases. However, we anticipate that incorporating a suction-based end-effector would help address these scenarios.

2.5.4 Servo Class Integration

In our previous work, we used a reactive servoing approach to follow the tracked bag in real time. This method relied on the Servo class from MoveIt Servo, which allowed the robot to continuously update its motion goal based on the live position of the object. The centroid of the plastic bag was published as it moved, and the arm subscribed to this data stream to follow the target. This approach was effective in scenarios with no drift.

In the current work, we adopt a different strategy by using the Mover class provided by MoveIt Servo. This interface enables the robot to move to a specific, predetermined point in space. We use this capability to direct the manipulator toward a predicted grasp point along the bag's trajectory. Rather than reacting to constant position updates, the system arrives at a position ahead of the bag and waits for the bag to reach the interception point. This approach improves controllability, especially in scenarios that require predicting future object positions.

2.5.5 Limitations of System

During testing, the detector struggled with a new test tank setup that introduced a black tarp background, which differed from the training environments. While useful for evaluating generalization, the tarp added unexpected challenges. Its glare and bag-like texture contrast sharply with the more consistent turbid water and metal tank backgrounds used during training.

False positives appeared near the frame edges when low-frequency disturbances occurred before the plastic bag entered view. This was partly expected, as the detector is expected to struggle with partial views, such as fragments in the corners. However, misclassifying background elements or motion artifacts as plastic bags points to a failure mode likely tied to the limited size of the training dataset. That said, this limitation may be less impactful in calmer, more realistic underwater environments, where turbid backgrounds are more uniform and vehicles can center bags within the camera frame.

Another limitation was observed in the velocity tracker. Since velocity is calculated based on the bag's initial descent values, it may underestimate the velocity that stabilizes later during the fall. This can lead to predictions of the bag's position that lag slightly behind its actual trajectory.

Finally, reachability is inherently constrained by the manipulator's physical range and the camera's field of view. Thus, limiting interaction to objects fully visible within the frame.

2.5.6 Future Works

There are several directions for future development. Expanding the existing FindingPlastic dataset is expected to improve model performance, particularly under the more challenging visual conditions observed during trials. During testing, the uniform tarp background introduced unexpected difficulty, as the model had not encountered similar textures during training. To address this, future data collection should include images with the tarp present, along with more edge cases such as partially visible bags or bags exiting the frame. Looking ahead to deployment, it will also be important to gather examples from natural aquatic settings, where background clutter and lighting vary more significantly than in controlled environments.

Within the reachability dataset, interpolation served as a reasonable approximation for planning purposes. However, more dense modeling or integration of real-time trajectory replanning may be necessary for future implementations.

Future work could also explore dynamic reachability limits in more depth. A preliminary 1-DOF analysis in the appendix (Figure B.1) offers a basic profile of vertical interception capabilities, but extending this to a full model for all positions would help identify failure regions and inform system tuning for more aggressive or edge-case trajectories.

While multi-object tracking is already supported, deciding how to prioritize and collect several items efficiently would make the system more capable in real-world deployments. Motion modeling should also be improved. The current planner assumes simple vertical paths and does not adjust during execution. Real underwater scenes would necessitate a 3D model of underwater dynamics. Integrating Kalman filters and leveraging the existing DeepSort tracking model would be a practical first step toward handling 3D motion dynamics underwater.

Completing the physical grasping process remains an important step. Adding a suction gripper attachment to the robotic arm would enable the system to collect flexible, drifting targets. This approach could make grasps more forgiving and more effective in uncertain underwater conditions. In addition, the system would benefit from basic feedback sensing to confirm successful grasps.

2.6 Summary

This chapter presents a vision-guided system for intercepting vertically drifting plastic bags using an underwater manipulator. Building on the FindingPlastic detection model, the system predicts future grasp points by modeling simplified vertical descent and evaluating reachability using a precomputed dataset. Tests in a simulated tank environment showed the system could anticipate and intercept drifting bags in real time. These results highlight the potential of the system for autonomous underwater collection of dynamic plastic bags.

Conclusion

The need to address plastic pollution, such as plastic bags, in marine ecosystems is growing. Once in the ocean, they can harm wildlife, break down into microplastics, and contribute to long-term environmental damage. Removing these materials before they degrade is more practical than addressing their long-term environmental impacts later.

This thesis presents the design and implementation of a vision-based autonomous robotic system that integrates perception, planning, and manipulation to address the challenge of marine plastic bag pollution.

In Chapter One, we introduced the FindingTrash detection model, which was trained on a curated dataset of underwater plastic bags. This detection system was combined with the DeepSORT tracking algorithm and an existing manipulation strategy, enabling real-time robot responses to observed targets. Chapter Two extended this capability by integrating the proven vision-based elements to address dynamically drifting plastic bags. Here, tracking was coupled with a new trajectory planner and motion method, allowing the system to predict future object positions and evaluate interception positions.

Together, these chapters build on each other to form an end-to-end pipeline for underwater plastic bag collection, progressing from initial feasibility to predictive, real-time interception of drifting objects. The modular design supports future enhancements, with components like detection, prediction, and manipulation easily updated for new objects, environments, or broader tasks such as sampling and retrieval. Looking ahead, we aim to deploy the system on an ROV or AUV as a practical solution for marine plastic bag remediation.

Overall, this thesis demonstrates the feasibility of autonomous underwater capture of plastic bags, marking an important step toward robotic solutions for marine pollution remediation.

Bibliography

[1] Anthony L. Andrady. "Microplastics in the Marine Environment". In: *Marine Pollution Bulletin* 62 (Aug. 2011), pp. 1596–1605. DOI: 10.1016/j.marpolbul.2011.05.030. URL: https://www.sciencedirect.com/science/article/pii/S0025326X11003055.

[2] Greg B Merrill et al. "Acoustic signature of plastic marine debris mimics the prey items of deep-diving cetaceans". In: *Marine Pollution Bulletin* 209 (Oct. 2024), pp. 117069–117069. DOI: 10.1016/j.marpolbul.2024.117069.

[3] Lauren Roman et al. "Plastic pollution is killing marine megafauna, but how do we prioritize policies to reduce mortality?" In: *Conservation Letters* 14 (Dec. 2020). DOI: 10.1111/conl.12781. URL: https://conbio.onlinelibrary.wiley.com/doi/10.1111/conl.12781.

[4] Luís Gabriel Antão Barboza et al. "Marine microplastic debris: An emerging issue for food security, food safety and human health". In: *Marine Pollution Bulletin* 133 (Aug. 2018), pp. 336–348. DOI: 10.1016/j.marpolbul.2018.05.047.

[5] Rachel Hurley et al. "Measuring riverine macroplastic: Methods, harmonisation, and quality control". In: *Water Research* 235 (May 2023), p. 119902. DOI: 10.1016/j.watres.2023.119902.

[6] Martina Fileš et al. "Analysis of the Mechanical Degradability of Biodegradable Polymer-Based Bags in Different Environments". In: *Sustainability* 16 (Mar. 2024), p. 2579. DOI: 10.3390/su16062579. URL: https://mdpi-res.com/sustainability/sustainability-16-02579/article_deploy/sustainability-16-02579.pdf?version=1711016912.

[7] López-Ibáñez S;Quade J;Wlodarczyk A;Abad MJ;Beiras R. "Marine degradation and ecotoxicity of conventional, recycled and compostable plastic bags". In: *Environmental pollution (Barking, Essex*

: 1987) 351 (2024). DOI: 10.1016/j.envpol.2024.124096. URL: https://pubmed.ncbi.nlm.nih.gov/38703982/.

[8] Christian T K.-H. "Marine Plastics: Innovative Solutions to Tackling Waste". In: *International Journal of Environmental Studies* (Feb. 2025), pp. 1–6. DOI: 10.1080/00207233.2025.2464473.

[9] Lauren Roman et al. "Plastic pollution in a rapidly developing nation: A comprehensive assessment of litter and marine debris surrounding coastal Cambodia". In: *Marine Pollution Bulletin* (Sept. 2024), pp. 116872–116872. DOI: 10.1016/j.marpolbul.2024.116872.

[10] Open Robotics. *ROS.org | Powering the world's robots*. Ros.org, 2020. URL: https://www.ros.org/.

[11] Michael Fulton et al. "Robotic Detection of Marine Litter Using Deep Visual Detection Models". In: *arXiv:1804.01079 [cs]* (Sept. 2018). URL: https://arxiv.org/abs/1804.01079.

[12] Glenn Jocher. *YOLOv8 Documentation*. docs.ultralytics.com, May 2020. URL: https://docs.ultralytics.com/.

[13] Harsh Panwar et al. "AquaVision: Automating the detection of waste in water bodies using deep transfer learning". In: *Case Studies in Chemical and Environmental Engineering* (July 2020), p. 100026. DOI: 10.1016/j.cscee.2020.100026.

[14] Pedro F. Proença and Pedro Simões. "TACO: Trash Annotations in Context for Litter Detection". In: *arXiv:2003.06975 [cs]* (Mar. 2020). URL: https://arxiv.org/abs/2003.06975.

[15] Gautam Tata et al. *A Robotic Approach towards Quantifying Epipelagic Bound Plastic Using Deep Visual Models*. URL: https://arxiv.org/pdf/2105.01882.pdf (visited on 04/14/2024).

[16] Gautam Tata. *gautamtata/DeepPlastic*. GitHub, Jan. 2022. URL: https://github.com/gautamtata/DeepPlastic.

[17] Glenn Jocher. *ultralytics/yolov5*. GitHub, Aug. 2020. URL: https://github.com/ultralytics/yolov5.

[18] Ultralytics. *YOLOv4*. docs.ultralytics.com. URL: https://docs.ultralytics.com/models/yolov4/.

[19] Muhammad Hussain. “YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection”. In: *Machines* 11 (July 2023), p. 677. DOI: 10.3390/machines11070677. URL: https://www.mdpi.com/2075-1702/11/7/677.

[20] Ghania Zidani et al. “OPTIMIZING PEDESTRIAN TRACKING FOR ROBUST PERCEPTION WITH YOLOv8 AND DEEPSORT”. In: *Applied Computer Science* 20 (Mar. 2024), pp. 72–84. DOI: 10.35784/acs-2024-05. (Visited on 06/04/2024).

[21] Mikel Broström. *BoxMOT: pluggable SOTA tracking modules for object detection, segmentation and pose estimation models*. GitHub, June 2024. URL: https://github.com/mikel-brostrom/boxmot.

[22] *Roboflow: Go from Raw Images to a Trained Computer Vision Model in Minutes.* roboflow.ai. URL: https://roboflow.com/.

[23] *Realtime Arm Servoing — MoveIt Documentation: Humble documentation*. Picknik.ai, 2025. URL: https://moveit.picknik.ai/humble/doc/examples/realtime_servo/realtime_servo_tutorial.html (visited on 05/20/2025).

[24] *Reach Bravo: Larger Platform ROV Manipulator*. REACH ROBOTICS. URL: https://reachrobotics.com/products/manipulators/reach-bravo/.

[25] Chris Wilcox et al. “Using expert elicitation to estimate the impacts of plastic pollution on marine wildlife”. In: *Marine Policy* 65 (Mar. 2016), pp. 107–114. DOI: 10.1016/j.marpol.2015.10.014. URL: https://www.sciencedirect.com/science/article/pii/S0308597X15002985.

[26] Tanmay Agarwal, Michael Fulton, and Junaed Sattar. “Predicting the Future Motion of Divers for Enhanced Underwater Human-Robot Collaboration”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Sept. 2021), pp. 5379–5386. DOI: 10.1109/iros51168.2021.9636374. URL: https://ieeexplore.ieee.org/abstract/document/9636374?casa_token=-4wVi0rtFVYAAAAA:Pi_r9SMlpiDzOHjIBTXrCeE8CyQoQ-EiL5yBgtw7K4mm2B8_Q7Err7ohnDgxt3febRyz2K-e (visited on 05/27/2025).

[27] Cong Wang et al. “Learning-Based Underwater Autonomous Grasping via 3D Point Cloud”. In: *OCEANS 2021: San Diego – Porto* (Sept. 2021), pp. 1–5. DOI: 10.23919/oceans44145.2021.9706035. URL: https://ieeexplore.ieee.org/abstract/document/9706035 (visited on 05/12/2025).

[28] Nicholas Guimond, Aaron Marburg, and Howard Chizeck. “FindingPlastic: Underwater Plastic Bag Detection and Retrieval”. In: *OCEANS 2024 - Halifax* (Sept. 2024), pp. 1–6. DOI: 10.1109/oceans55160.2024.10753930. URL: https://ieeexplore.ieee.org/document/10753930 (visited on 05/12/2025).

[29] Naresh Marturi et al. “Dynamic grasp and trajectory planning for moving objects”. In: *Autonomous Robots* 43 (June 2019), pp. 1241–1256. DOI: 10.1007/s10514-018-9799-1. URL: https://link.springer.com/article/10.1007%2Fs10514-018-9799-1.

[30] Baichuan Huang, Jingjin Yu, and Siddarth Jain. *EARL: Eye-on-Hand Reinforcement Learner for Dynamic Grasping with Active Pose Estimation*. arXiv.org, 2023. URL: https://arxiv.org/abs/2310.06751 (visited on 05/12/2025).

Chapter A

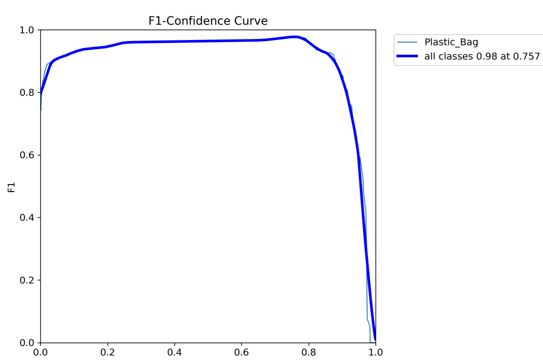
Appendix for Chapter I.

Detection Model Performance Curves

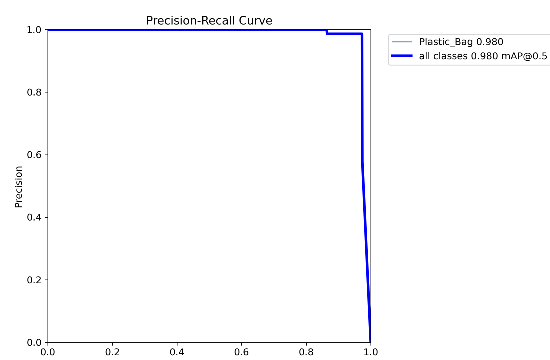
Figure A.1 shows evaluation curves for the FindingPlastic detection model, which was trained to identify a single class: Plastic_Bag.

On the left, the F1-confidence curve plots the F1 score against detection confidence thresholds. The peak F1 score of 0.98 occurs at a confidence level of 0.757, reflecting the best balance between precision and recall. Scores drop off at higher confidence thresholds due to stricter filtering of detections.

On the right, the precision-recall curve illustrates that the model keeps precision near 0.98 across most recall levels, indicating strong and consistent detection. The mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) is 0.980 for the Plastic_Bag class.



(a) F1-Confidence Curve



(b) Precision-Recall Curve

Figure A.1: F1-Confidence and Precision-Recall curves from FindingPlastic.

The normalized confusion matrix (Figure A.2) highlights the performance of the trained model on two classes: Plastic_Bag and background. These were the only classes used in the detection task. The matrix shows a high true positive rate for Plastic_Bag detections, with minimal misclassification into the background class.

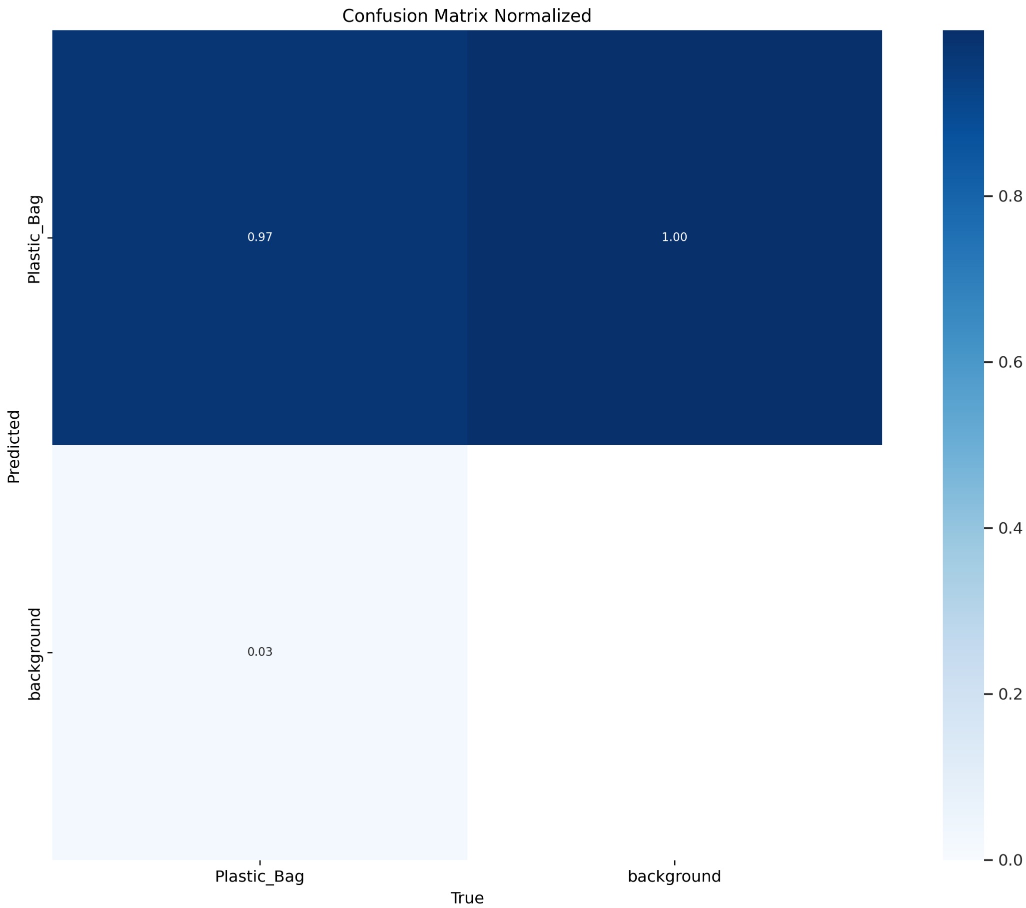


Figure A.2: Normalized confusion matrix showing classification performance between the two classes: Plastic_Bag and background.

Chapter B

Appendix for Chapter II.

Vertical Interception Feasibility Model (Preliminary Analysis)

To test the limits of vertical interception, we implemented a simplified 1-DOF reachability model. In this setup, the plastic bag's lateral position was fixed at (x_0, y_0) with no horizontal drift ($v_x = v_y = 0$), and only the vertical velocity v_z was varied. For any configuration in the workspace, in this case (Figure B.1), fixed values of $x = 0.50$ m and $y = -0.40$ m, a sweep is performed over a specified range of vertical velocities v_z. The objective is to assess the feasibility of a successful catch at these positions.

The plot shown illustrates a binary feasibility outcome (1 = Yes, 0 = No) as a function of v_z, indicating whether a catch is dynamically achievable given the initial spatial configuration and varying only the vertical speed component. In this particular case, the catch is feasible for vertical velocities up to approximately 0.09 m/s, beyond which the feasibility condition fails. This setup enables the systematic exploration of feasible velocity bounds across spatial points of interest.

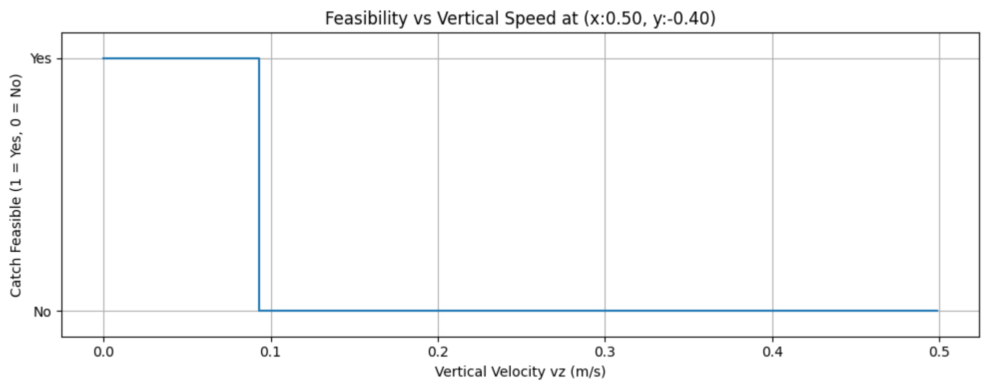


Figure B.1: Estimated maximum vertical velocity $v_{z,\max}$ that the system can intercept at different lateral positions x_0, under a simplified 1-DOF model assuming no lateral drift. This provides an initial characterization of spatial reachability limits.

The analysis would then be repeated across several lateral positions x_0 within the robot's workspace, generating a curve of maximum interceptable vertical velocities $v_{z,\max}(x_0)$.

This model is preliminary and does not capture the full dynamics or grasping constraints of the real system. However, it serves as a first-order estimate of vertical interception capability. If expanded in future work, a more comprehensive version of this analysis could help define the performance envelope of the system more rigorously and guide the design of motion strategies near the limits of reachability.