**Software Documentation**

**1. Introduction**

**1.1 Project Overview**

This project aims to develop an **autonomous underwater garbage detection and retrieval system** using **YOLOv8** for object detection, **Intel RealSense** for depth estimation, and an **Arduino-controlled robotic arm** for object pickup. The system identifies underwater waste, determines its 3D location, and actuates a robotic arm to retrieve the detected objects.

**1.2 Objectives**

* Implement **YOLOv8** for real-time detection of underwater garbage.
* Use **Intel RealSense depth camera** to extract depth information.
* Develop **inverse kinematics (IK) calculations** for robotic arm control.
* Integrate an **Arduino-based robotic arm** to pick up detected waste.

**2. System Components & Specifications**

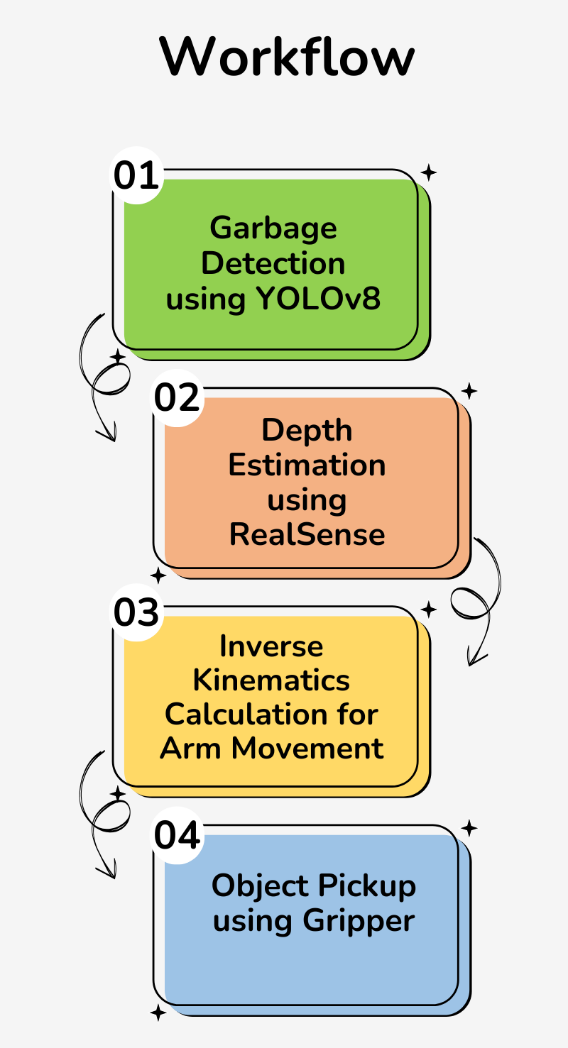
|  |  |
| --- | --- |
| **Camera** | **Intel RealSense** |
| **Microcontroller** | **Arduino Uno** |
| **Robotic Arm** | **3 DOF Servo-based Arm** |
| **Embedded Processing** | **Jetson Nano** |
| **Power Supply** | **Battery** |

**2.1 Hardware Components**

**2.2 Software & Libraries**

|  |  |
| --- | --- |
| **YOLOv8** | **Garbage detection** |
| **PyRealSense2** | **Depth estimation** |
| **OpenCV** | **Image processing** |
| **NumPy & SciPy** | **Mathematical computations** |
| **Serial Communication** | **Arduino control** |
|  |  |

**3. Four-step workflow**



**4. Implementation**

**4.1 Garbage Detection using YOLOv8**

* **Dataset**: Underwater images dataset (<https://universe.roboflow.com/neural-ocean/neural_ocean/dataset/3>)
* **Model Training**: YOLOv8 is trained on annotated underwater waste images.
* **Detection Output**: Bounding boxes (x\_min, y\_min, x\_max, y\_max).

**Output**: Bounding box coordinates for detected garbage.

**4.2 Depth Estimation using RealSense**

* Extract depth from the bounding box **center (x, y)** using RealSense depth data.
* Convert **2D coordinates to 3D world coordinates (X, Y, Z)**.

**Output**: Real-world **(X, Y, Z) position** of the detected garbage.

**4.3 Inverse Kinematics Calculation**

* Convert **(X, Y, Z) position** to **joint angles** for the robotic arm.
* Implement **manual inverse kinematics (IK)** calculations.

**Output**: Servo angles **(θ1, θ2, θ3)** for robotic arm movement.

**4.4 Robotic Arm Control & Object Pickup**

* Send computed **servo angles** to **Arduino** via serial communication.
* Control the **gripper** to pick up the garbage.

**Output**: Arm moves to the detected object and picks it up.

**Submitter:  
Nikhil Singh Shekhawat (4 March 2025)**