Autonomous Trash-Skimming Boat Using Lightweight YOLO

Muhammad Abdullah [1] **Zaeem Mohtashim Khan** [1]

Abstract

Floating waste in urban waterways poses a significant environmental challenge. This project explores the development of an autonomous trash-skimming boat that uses computer vision to detect and navigate toward debris. We retrofit a remote-controlled (RC) boat with a camera, a Jetson Nano (2GB), and an Arduino for motor control. A lightweight YOLO model is used for real-time object detection, enabling the boat to identify floating trash and respond with directional movement. While our system currently performs inference offboard on a laptop due to Jetson Nano memory limitations, the architecture is designed with future onboard deployment in mind. Initial results demonstrate reliable detection of floating waste, offering a low-cost and modular approach to automated water surface cleaning.

Code and resources are available at: `https://github.com/Abdullah-Aseeef/trash_skimming_boat`

1. Introduction

The accumulation of floating trash in water bodies has become an escalating environmental issue, particularly in urban areas where waste disposal systems are often overwhelmed. Manual methods of trash collection are labor-intensive, inefficient, and difficult to scale, especially in hard-to-reach or hazardous locations. Autonomous systems offer a promising alternative for continuous, low-cost, and scalable waterway cleanup.

Recent advances in computer vision and embedded systems have enabled the use of lightweight object detection models for mobile robotic platforms. However, deploying such models in resource-constrained environments—such as on small-scale boats operating in outdoor conditions—poses unique challenges, including computational limitations, real-time responsiveness, and robust detection under variable lighting and environmental noise.

In this paper, we present an autonomous trash-skimming boat that uses a real-time object detection pipeline powered by a lightweight YOLOv12s model. Our system retrofits a remote-controlled (RC) boat with a camera, an NVIDIA Jetson Nano, and an Arduino microcontroller. Due to the memory and compatibility limitations of the 2GB Jetson Nano, we adopt a hybrid processing architecture in which the Jetson captures frames and communicates with an off-board laptop that performs inference. Based on the detection results, the laptop issues movement commands that are relayed back to the boat via the Jetson and then executed by the Arduino to actuate the motors.

We describe our iterative training process, starting with a multi-class model trained on the *Waste in Water* dataset and progressing to a high-performing single-class model trained on the *Trash in Water Channels* dataset. The final model achieves a substantial improvement in validation mAP—from 21.1 to 80.0—and performs robustly on unseen videos from the *Flow_Img* dataset. Our contributions are twofold: (1) we propose a practical, low-cost autonomous system for floating trash detection and navigation; and (2) we evaluate the effectiveness of dataset selection and training strategies for deploying lightweight detection models in constrained edge computing environments.

2. Related Work

Trash detection in aquatic environments has gained significant attention due to the increasing need for scalable and automated environmental cleanup solutions. Several studies have explored object detection models—particularly the YOLO family—for identifying and locating trash in challenging conditions.

Trash Detection in Water Bodies Using YOLO with Explainable AI Insight (Rasheed et al., 2024)
Engineers from the University of Engineering and Technology Taxila and the National Center of Robotics and Automation evaluated YOLOv5, YOLOv6, YOLOv7, YOLO-NAS, and YOLOv9 across four datasets: Trash Flow Rate Dataset, DeepTrash, FloW, and TrashICRA. Their experiments demonstrated that YOLOv9 (GELANC) achieved the highest mean average precision (mAP), particularly in low-visibility and noisy environments. They also employed Grad-CAM to visualize model attention, offering interpretability. This work provides a valuable baseline for our model selection and performance expectations on edge devices.

OATCR: Outdoor Autonomous Trash-Collecting Robot Design Using YOLOv4-Tiny (Kulshreshtha et al., 2021)
Kulshreshtha et al. designed an autonomous land robot that detects and collects trash using YOLOv4-Tiny, achieving 95.2% mAP with an inference time of 5.21 ms. Their system integrated GPS and onboard sensors for navigation. While our boat operates in water, their efficient real-time detection pipeline and lightweight model selection are highly relevant for embedded deployments like ours.

Autonomous Trash Collector Based on Object Detection Using Deep Neural Network (Hossain et al., 2019)
This study proposed a trash collection robot powered by a Raspberry Pi and a custom CNN. An ultrasonic sensor was used to trigger image capture only when an object was within range, optimizing resource usage. Though the model was built from scratch, the paper shows the feasibility of deploying deep learning models on low-power devices and demonstrates the value of intelligent sensor fusion—an idea that can be extended to our water-based system.

Trash Detection on Water Channels (Tharani et al., 2020)
This work focuses on identifying floating trash in complex drainage systems. The authors curated a dataset of 13,500 images with 48,450 annotated instances and evaluated YOLO variants, RetinaNet, and PeleeNet. They introduced enhancements such as log-attention, depth-wise separable convolutions, and focal loss to improve detection of occluded and partially submerged trash. Their insights into reflections, variable object sizes, and class imbalance inform the challenges we address in water-based object detection.

Machine Learning-enabled Robotic Trash Collector (Choudhary et al., 2024)
The MLRTC project developed a floating robotic platform using YOLOv3 for detection, a Bluetooth module for communication, and a webcam for real-time monitoring. Built on lightweight materials and solar power, the system offers a cost-effective and sustainable solution. Despite limitations like inconsistent lighting and environmental instability, the project demonstrates key principles in float-based detection and actuation that align closely with our boat-based design.

FloW: A Dataset and Benchmark for Floating Waste Detection (Cheng et al., 2021):
Cheng et al. introduced the FloW dataset, a high-resolution video dataset designed to benchmark object detection models for floating waste in real-world river and stream environments. The dataset comprises more than 7,000 annotated frames across diverse scenarios and includes challenging conditions such as glare, occlusion, water turbulence, and cluttered backgrounds. The authors provide benchmark results using Faster R-CNN, RetinaNet, and YOLOv4, and emphasize the importance of spatial-temporal consistency in evaluating detection performance in video streams. This dataset forms a key part of our evaluation, as our improved model was tested on its video and image subset (*Flow_Img*) to assess real-world generalization.

3. Datasets Used

Our project relies on three key datasets, each serving a distinct purpose in the model training and evaluation pipeline:

1. Waste in Water Dataset (Chinatele, 2024)
This dataset was used for the initial baseline training. It contains labeled images of waste floating in or near water bodies, formatted in YOLO annotation style. However, the dataset includes several images captured from land or unrelated contexts, which contributed to poor generalization in water-focused detection. Despite this, it provided a useful starting point for architecture validation and early benchmarks.

2. Trash in Water Channels (Tharani et al., 2020)
This dataset forms the core of our final training phase. It comprises 13,500 images with 48,450 annotated instances of floating trash across various drainage channels and urban water systems. The dataset is rich in visual diversity, including scenes with occlusion, glare, and background noise. We converted the original annotation format into YOLO-style using a custom Python pipeline and trained our model for 100 epochs on this dataset. It significantly improved detection accuracy and robustness.

3. Flow_Img Subset from FloW Dataset (Cheng et al., 2021)
To test real-world generalization, we used the Flow_Img subset from the FloW benchmark dataset. This subset contains high-resolution video frames captured onboard boats in Chinese rivers and canals. It features realistic water dynamics, floating debris, and environmental variability. The dataset was not used for training and served solely as a test set to assess the model's ability to perform under operational deployment conditions.

4. Methodology

4.1. Model and Training

We used the YOLOv12s model as the foundation for our object detection system. For our baseline, the model was trained for 50 epochs on the *Waste in Water* dataset (Chinatele, 2024), a publicly available dataset containing annotated images of waterborne waste. This dataset, while readily formatted for YOLO training, included a mix of image types, including some with land-based trash, which limited its relevance for our specific use case focused solely on floating trash in water.

To improve performance, we increased the number of training epochs to 100 and reformulated the task from a multi-class problem to a single-class detection problem. However, these adjustments did not yield a significant boost in performance. We hypothesized that this was due to the poor quality and inconsistency of the original dataset—many of the images were either taken from non-water contexts or had poor resolution and lighting, which confused the model.

In response, we transitioned to a more domain-specific dataset: *Trash in Water Channels* (Tharani et al., 2020). This dataset, introduced by Tharani et al., comprises 13,500 high-quality images with 48,450 annotated instances of floating trash captured in realistic urban drainage conditions. The dataset accounts for various challenges typical to real-world aquatic environments, such as reflections, water turbulence, trash clustering, and partial occlusion. Its diversity and robustness made it significantly more suitable for our task.

Training on this new dataset led to a substantial increase in model performance. When evaluated on unseen test footage from the *Flow_Img* dataset, we observed a dramatic improvement in detection of trash - however at the same time we saw more false positives as items outside the water body were also being flagged as waste. To counter this we used a rectangular mask to block off the part of the frame that does not contain water to allow model to focus only on the trash inside the water body.

4.2. Hardware Setup

The hardware pipeline for our autonomous system began with the NVIDIA Jetson Nano 2GB, intended to perform on-board inference for the YOLOv12s model. Initial deployment attempts used two different methods: native Yolo inference and TensorRT acceleration.

1. Native YOLO Inference: We first attempted to run YOLO directly on the Jetson using PyTorch. Despite the model consuming only 150–200 MB of RAM, we encountered compatibility issues—YOLO required Python 3.10 to run , while the Jetson was constrained to Python 3.6 due to legacy CUDA and PyTorch dependencies. Although we were able to run Python 3.10, Pytorch with Cuda could not be upgraded to support this version since it was linked with the legacy version which came with the hardware. The inability to upgrade PyTorch to a CUDA-enabled version for Python 3.10 resulted in the model running exclusively on CPU. As a result, inference speed dropped to approximately 4 seconds per frame, rendering real-time deployment infeasible.

2. TensorRT Acceleration: We then explored TensorRT-based acceleration to utilize the Jetson's GPU. However, this approach consumed around 1.6 GB of RAM, exceeding the Jetson's 2 GB memory limit. This led to aggressive swapping and throttling, further degrading performance to roughly 6 seconds per frame. These constraints made it clear that on-board inference was not viable within the available hardware limits.

3. Offboard Inference via Laptop: To circumvent these limitations, we offloaded the inference pipeline to a more capable laptop. The Jetson Nano was equipped with a MIPI CSI camera to capture frames. These frames were transmitted via socket communication to the laptop, which ran the YOLOv12s model. The model would detect trash and return movement commands based on the location of the largest bounding box—typically corresponding to the closest detected object. These directional commands (left, right, or forward) were sent back to the Jetson over the same socket interface.

4. Control Relay to Arduino: The Jetson, upon receiving commands from the laptop, relayed them to an Arduino via a serial connection. The Arduino was responsible for interfacing with the motor drivers and actuating the boat's motion according to the received direction.

This distributed system design allowed us to maintain real-time responsiveness while accommodating the resource limitations of the Jetson Nano.

5. Results

5.1. Model Evaluation on Benchmark Dataset

To evaluate the effectiveness of our model training strategy, we tested the final YOLOv12s model on the *Trash in Water Channels* dataset. This dataset is well-suited for benchmarking due to its diversity of scenes, lighting conditions, and realistic representations of floating trash in urban water bodies.

After iteratively improving the model—by increasing training epochs, shifting to a single-class classification problem, and switching from the *Waste in Water* dataset to *Trash in Water Channels*—we observed a significant boost in detection performance. The final trained model achieved the following evaluation metrics on the validation set:

- **Accuracy:** 0.7796
- **Precision:** 0.8748
- **Recall:** 0.8775
- **F1 Score:** 0.8762

These results demonstrate that the dataset change, combined with architectural and training adjustments, contributed meaningfully to the model's ability to generalize across various trash types and visual conditions.

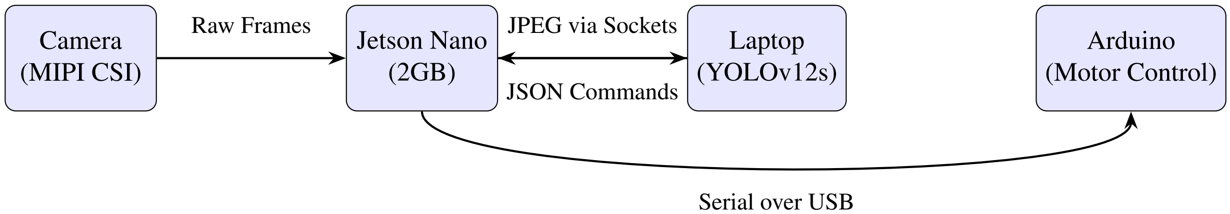

Figure 1. System architecture: pipeline from image capture to motor control.

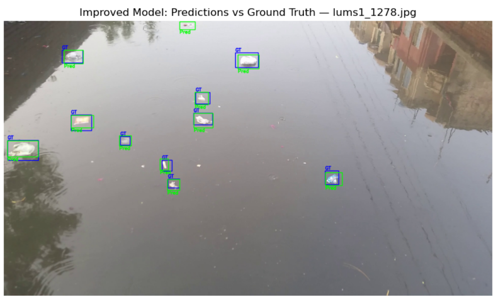

Figure 2. Sample Output from validation split of"Trash in Water Channel"

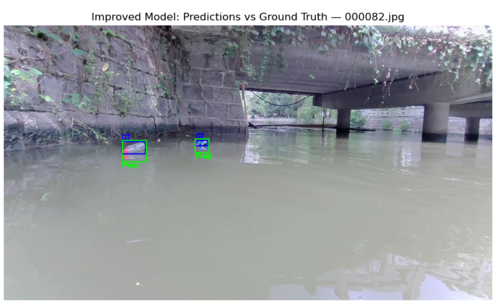

Figure 3. Sample Output from Flow_Img Dataset.

5.2. Real-World Performance on Unseen Data

To validate generalization in real-world conditions, we deployed the final model on unseen video data from the *Flow_Img* dataset. These videos, filmed onboard a boat operating in Chinese water bodies by the OrCA organization, represent dynamic aquatic environments with realistic motion and background noise.

The model demonstrated strong qualitative performance, successfully detecting and persistently tracking floating waste items across multiple frames. Compared to the baseline model, the improved version showed enhanced spatial consistency, fewer false positives, and better responsiveness to partial occlusions or lighting variations. This confirms the suitability of our system for deployment in operational scenarios and highlights the effectiveness of training on domain-relevant, high-quality datasets.

5.3. Hardware-Side Integration and Control

Beyond software performance, a critical aspect of this project was integrating the detection output with real-time motor control on the physical boat. After successfully offloading inference to a laptop and establishing bidirectional communication via socket programming, we implemented a control strategy to translate detection results into physical movement.

The system processed each incoming video frame to identify the location of the largest bounding box—interpreted as the closest visible piece of trash. Based on the horizontal position of this bounding box within the frame, the system issued movement commands:

- **Left:** if the object was detected in the left third of the frame
- **Right:** if the object appeared in the right third
- **Forward:** if the object was near the center

These commands were transmitted back to the Jetson Nano, which relayed them via serial communication to the onboard Arduino. The Arduino then powered the corresponding motors to actuate the boat in the appropriate direction.

Inference on the laptop, which was equipped with an RTX 3070 GPU and CUDA support, achieved runtimes in the range of tens of milliseconds per frame, enabling real-time performance at over 10 frames per second.

Although we were unable to test the system in an actual aquatic environment due to logistical constraints, controlled lab tests on a stationary setup confirmed that the detection-to-actuation loop functioned correctly. The motors responded reliably to the object's position in the video frame, validating our end-to-end pipeline from perception to control.

These results lay the groundwork for future real-world deployment, where this control logic can be refined further using feedback mechanisms such as IMUs, GPS, or visual servoing to improve path stability, object re-acquisition, and obstacle avoidance. Our modular design ensures that minimal changes will be required to transition the system from dry-run testing to in-field operation.

6. Discussion

6.1. Model Performance and Dataset Impact

The transition from the *Waste in Water* dataset to the *Trash in Water Channels* dataset had a dramatic effect on model performance, improving validation mAP from 21.1 to 80.0. This highlights the critical role of dataset quality and relevance in training effective object detection models for real-world applications. Our experience reinforces that even lightweight models like YOLOv12s can perform reliably when trained on task-specific, well-annotated data.

6.2. Hardware Limitations and Design Trade-offs

The 2GB Jetson Nano's limitations shaped much of our system design. While we initially aimed for onboard inference, the lack of CUDA support on Python 3.6 and memory bottlenecks during TensorRT execution forced us to offload processing to a laptop. This introduced a new set of challenges, including network latency and synchronization between devices. However, it allowed for real-time inference and maintained system responsiveness. A future iteration using a Jetson Xavier NX or Orin Nano could re-enable full edge deployment.

6.3. Model Limitations

Despite the promising performance of our system, several limitations remain, particularly on the model side. The most frequent issue observed is the misclassification of reflections—bright areas on the water surface, such as sunlight or glare, are often incorrectly detected as trash. Additionally, objects outside the water body, such as debris on sidewalks or leaves on a tree, are sometimes flagged as floating waste due to a lack of contextual awareness in the current model. Tracking also remains a challenge; since the system relies on selecting the largest bounding box per frame without persistent object tracking, it can result in jittery or inconsistent control, especially when multiple trash items briefly appear. These limitations motivate many of the future improvements we propose, including dynamic masking, reflection-aware preprocessing, and attention mechanisms to improve robustness and reliability in real-world deployments.

6.4. Communication Pipeline Reliability

Using socket communication between the Jetson and the laptop proved to be an efficient workaround, but it is inherently vulnerable to packet loss, bandwidth limitations, and WiFi reliability—especially in outdoor or water-based environments. Secondly, it is not always feasible to have the boat connected to WiFi - which further leads to running the model directly on the Jetson.

6.5. Control and Navigation Simplification

Our navigation logic is based on selecting the largest bounding box as a proxy for proximity. While this heuristic worked in controlled settings, it assumes a relatively clutter-free environment and consistent object sizes. In the future, integrating additional sensors (e.g., depth cameras or LiDAR) or combining distance from centre and distance form object for motor speed control will ensure better real life use cases.

6.6. Scalability and Environmental Impact

Although our prototype was tested in a limited environment, the underlying architecture—modular and low-cost—can scale to multiple boats for collaborative trash collection. Additionally, deploying solar-powered variants (as seen in related work) could significantly enhance functionality and reduce operational costs - especially in Pakistan where such approaches are seen as a novelty yet are in high need for the dire situation of our local water channels.

7. Future Directions

While our current system demonstrates the feasibility of real-time trash detection and reactive navigation, several enhancements are planned to improve robustness and autonomy in real-world scenarios.

1. Advanced Motor and Speed Control: Future iterations will include fine-grained control over motor speed and turning rates. This will allow the boat to adapt its movement based on the size and distance of detected objects, enabling smoother and more energy-efficient navigation.

2. Onboard Inference on Upgraded Hardware: Our current hardware limitations with the 2GB Jetson Nano restricted us to offboard inference. We plan to migrate the model to a 4GB Jetson Nano or equivalent, which would allow us to run the model fully onboard using TensorRT acceleration. This shift would eliminate network latency, reduce reliance on external computation, and make the system more portable and scalable. We can also attempt to use Yolo V12N which is one-third the size of the model we are using and run that either on a 2gb or 4gb Jetson if the current model size remains a bottleneck in running local inference.

3. Autonomous Search Algorithms: An important next step is designing search behavior for when no trash is detected. This logic will be adaptive based on the geometry of the water body:

- *Circular patterns* for lakes and enclosed basins
- *Sweeping back-and-forth* motions for canals or linear water flows

- *Spiral or radial expansion* in stagnant or limited-visibility environments

This behavior will ensure full area coverage and reduce idle time, improving the effectiveness of autonomous operation.

Together, these improvements will significantly increase the system's autonomy and make it viable for long-term deployment in a variety of urban and semi-natural water bodies.

4. Model Improvements for Visual Robustness: To further improve detection quality, we plan to implement:

- **Dynamic masking** that adapts in real-time based on water boundaries instead of relying on fixed frame regions.
- **Reflection reduction** via image preprocessing techniques (e.g., polarization filters or histogram equalization) to mitigate the impact of water surface glare.
- **Attention mechanisms** within the detection architecture to help the model focus on the most relevant regions of the image, particularly in cluttered scenes.

8. Conclusion

In this work, we present an autonomous trash-skimming boat system that combines real-time object detection with embedded hardware control to address the growing problem of floating waste in urban water bodies. By leveraging a lightweight YOLOv12s model and transitioning to a more domain-specific dataset, we significantly improved detection performance, achieving an F1 score of 0.8762. Due to the computational limitations of the 2GB Jetson Nano, we implemented a hybrid processing architecture where inference was offloaded to a laptop while retaining real-time responsiveness through socket-based communication. Our system successfully translated detection results into directional motor control via an Arduino interface and demonstrated functional closed-loop behavior in controlled tests. This project provides a low-cost, modular foundation for future autonomous surface vehicles and sets the stage for further improvements in onboard inference, adaptive navigation, and scalable deployment in diverse aquatic environments.

9. Contributions

This project makes the following key contributions:

- We develop a low-cost, modular hardware-software pipeline to enable autonomous trash detection and navigation using a retrofitted RC boat.
- We demonstrate how lightweight YOLOv12s can be adapted for real-time trash detection through careful dataset selection and architecture tuning, achieving an F1 score of 0.8762.
- We identify and address the limitations of deploying object detection models on a 2GB Jetson Nano, providing a hybrid offboard-inference solution via socket communication.
- We design and test a real-time control loop that translates detection results into motor commands, validated via physical actuation through Arduino in lab conditions.
- We propose a scalable architecture and outline future work to enable full onboard inference, adaptive navigation strategies, and long-term deployment in various water environments.

References

Cheng, Y., Zhu, J., Jiang, M., Fu, J., Pang, C., Wang, P., Sankaran, K., Onabola, O., Liu, Y., Liu, D., and Bengio, Y. Flow: A dataset and benchmark for floating waste detection in inland waters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10953–10962, October 2021.

Chinatele. waste in water dataset. `https://universe.roboflow.com/chinatele/waste-in-water`, jan 2024. URL `https://universe.roboflow.com/chinatele/waste-in-water`. visited on 2025-05-06.

Choudhary, R., Jadhav, R., Rokde, H., Kalaskar, R., Rokade, B., and Rupnavar, S. Machine learning-enabled robotic trash collector. In *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, pp. 157–163, 2024. doi: 10.1109/ICSES63445.2024.10762994.

Hossain, S., Debnath, B., Anika, A., Junaed-Al-Hossain, M., Biswas, S., and Shahnaz, C. Autonomous trash collector based on object detection using deep neural network. In *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pp. 1406–1410, 2019. doi: 10.1109/TENCON.2019.8929270.

Kulshreshtha, M., Chandra, S., Randhawa, P., Tsaramirsis, G., Khadidos, A., and Khadidos, A. Oatcr: Outdoor autonomous trash-collecting robot design using yolov4-tiny. *Electronics*, 10:2292, 09 2021. doi: 10.3390/electronics10182292.

Rasheed, S., Mirza, A., Saeed, M. S., and Yousaf, M. H. Trash detection in water bodies using yolo with explainable ai insight. In *2024 International Conference on*

Robotics and Automation in Industry (ICRAI), pp. 1–7, 2024. doi: 10.1109/ICRAI62391.2024.10894205.

Tharani, M., Amin, A. W., Maaz, M., and Taj, M. Attention neural network for trash detection on water channels, 2020. URL https://arxiv.org/abs/2007.04639.