ISSN: 2996-2625

Research Article

Medicine and Pharmacology: Open Access

Surface Water Garbage Detection, Classification and Counting Using Modern Image Recognition

Prathamesh Garud*

Sanjivani University, Kopargaon, Ahilyanagar, Maharashtra, India

***Corresponding Author**
Prathamesh Garud, Sanjivani University, Kopargaon, Ahilyanagar, Maharashtra, India

Submitted: 2026, Oct 15; **Accepted:** 2026, Nov 10; **Published:** 2026, Nov 13

Citation: Garud, P. (2026). Surface Water Garbage Detection, Classification and Counting Using Modern Image Recognition. *Med Pharmacol OA, 4*(2), 01-05.

Abstract

Garbage floating on rivers, lakes and ponds harms water life, blocks drains and spoils the look of public places. Checking it by eye is slow and tiring. This paper builds a simple image-based system that finds floating garbage in a photo, tells what type it is, and counts how many pieces of each type are present. A small YOLOv8 model (YOLOv8n, about 3.0 million parameters) was trained on a public water-garbage image set with 371 training images, 105 validation images and 7 labels. Counting is done by adding up the boxes the model draws for each class. In the training log used for this paper (24 completed epochs), the model reached a best mAP50 of 0.244, with precision of 0.624 and recall of 0.276 at that point, and it took about 155 ms to process one image on a CPU. These scores are modest. They show that the idea works even with very little data and no GPU, and they point clearly to what must improve: more labelled images, cleaner class labels, and a longer training run.

Keywords: Water Pollution, Floating Garbage, Object Detection, Yolov8, Image Classification, Object Counting, Deep Learning

1. Introduction

Plastic and other waste reach rivers, lakes and canals every day. Once in the water, it can trap fish and birds, clog drains and break into smaller pieces that are very hard to remove. A well-known estimate by Jam beck et al. showed how large the flow of plastic into the sea is, which makes regular monitoring of inland waters important [1].

Today most monitoring is done by people walking along the bank or travelling by boat and writing down what they see. This is slow, differs from person to person and cannot be repeated often. A camera on a bridge, a boat or a drone could take the same pictures every day, if a computer could read them. Reading means three jobs: finding each piece of garbage (detection), saying what kind it is (classification) and saying how many there are (counting).

Modern object-detection networks can do the first two jobs in one step, and the third follows naturally once every item has its own box. This study uses one such network, YOLOv8, on a small public data set and asks a practical question: how well can a small model, trained on a normal laptop-class CPU, find and count floating garbage? The aims are (i) to train a detector on water-garbage images, (ii) to measure it with standard detection scores, (iii) to show how counting is obtained from the detections, and (iv) to list honestly what limits the result.

Section II reviews earlier work, Section III explains the data and method, Section IV gives the results, and Section V closes with limits and future work.

2. Related Work

Table I lists the main studies and tools that this work builds on. Detection networks fall into two families. Two-stage models such as Faster R-CNN first suggest regions and then classify them; they are accurate but slower [3]. One-stage models such as YOLO and SSD look at the whole picture once and are quick enough for live video [2,4]. Fulton et al. compared several of these on underwater plastic litter and showed that deep detectors can work on this kind of scene [5]. Land-based litter has the TACO collection, and floating waste on inland water has the FloW benchmark [6,7]. YOLOv8 is a recent, easy-to-use release of the YOLO family [8].

#	Study	Contribution	Open point
1	Jambeck et al. [1]	Estimated plastic entering the sea from land	Needs cheap, repeatable monitoring
2	Redmon et al. [2]	YOLO: one-pass detection	Early version, weak on small items
3	Ren et al. [3]	Faster R-CNN: two-stage detector	Slower for live use
4	Liu et al. [4]	SSD: one-pass, many scales	Not built for waste
5	Fulton et al. [5]	Detectors for underwater litter	Underwater, not surface water
6	Proença and Simões [6]	TACO litter data set	Litter on land
7	Cheng et al. [7]	FloW floating-waste benchmark	Only a few object types
8	Jocher et al. [8]	YOLOv8 software	General purpose, not tuned for waste

Table 1: Selected Related Work

Taken together, the earlier work shows that the detectors exist and that litter data sets exist, but few small, low-cost studies report both the type and the number of garbage items on the water surface. This paper tries to fill that gap in a modest way.

3. Methodology

A. Workflow

Figure. 1 shows the steps. The labelled images are used to fine-tune a YOLOv8n model that was already trained on a large general data set. After each epoch the model is checked on the validation images. Once trained, the model is given a new photo, draws a box and a label around every item it finds, and the boxes are counted per class.

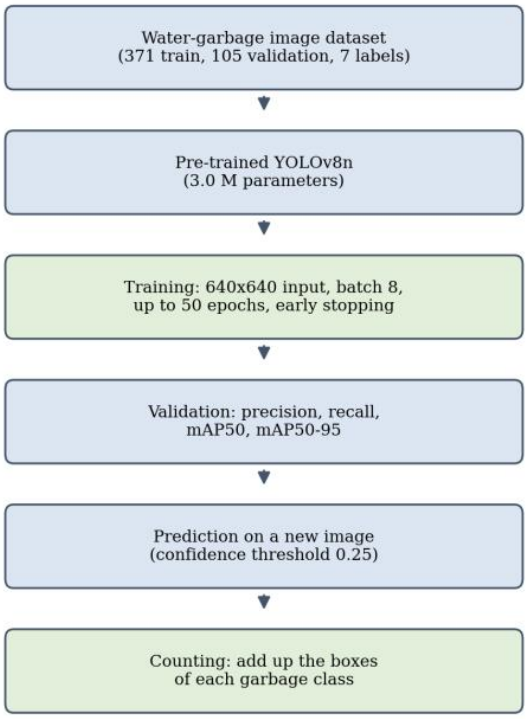


Figure 1: Workflow of the Proposed System

B. Dataset

The images come from a public Roboflow Universe project called Object-Detection (version 1, licence CC BY 4.0), exported in YOLOv8 format [9]. The split used here has 371 training images and 105 validation images; a separate test folder is also supplied but was not used in the run reported here. The 105 validation images contain 565 labelled objects, which is about five objects per image. The data file lists seven labels: No object detection, Object-Detection, Plastic Bag, Plastic Bottle, Plastic Can, Waste fibre and other waste. The first two look like placeholder names left by the labelling tool rather than real garbage types, and the original file spells some names differently (for example “Bottel”

and “faiber”). They are kept as they are so the results match the data file. Per-class scores were not part of the log, so the reported numbers are averages over all classes.

C. Model and Training Setup

YOLOv8n is the smallest model of the YOLOv8 family. It has 130 layers, 3,012,213 parameters and needs 8.2 GFLOPs per image. Training used the Ultralytics library with the settings in Table II. All other settings were left at the library defaults, including the default image augmentation (mosaic, flips and colour changes). Training ran on a CPU, taking about 3.5 minutes per epoch.

Setting	Value
Starting weights	yolov8n.pt (pre-trained)
Image size	640 x 640
Batch size	8
Planned epochs	50
Early stopping patience	10 epochs
Hardware	CPU only

Table 2: Training Settings

D. Counting

No separate counting network is needed. For one image, the model returns a list of boxes, each with a class and a confidence value. Boxes with confidence below 0.25 are dropped. The remaining boxes are grouped by class, and the size of each group is the count for that class. For example, three boxes labelled Plastic Bottle give a count of three bottles.

E. Evaluation Measures

A box is counted as correct when it overlaps a true box enough, measured by intersection over union (IoU). Precision is TP/(TP+FP), the share of drawn boxes that are right. Recall is TP/(TP+FN), the share of real objects that were found. Average precision (AP) summarises precision and recall for one class, and mAP is its mean over classes. mAP50 uses an IoU limit of 0.5, while mAP50-95 averages the limits from 0.5 to 0.95 and is stricter [10]. Because counting depends on missed and extra boxes, recall and precision matter directly for count quality.

4. Results and Discussion

A. Training Progress

The log used here covers 24 completed epochs; the run was cut off during epoch 25, before the 50 planned epochs or the early-stopping rule ended it. Scores were still rising slowly, so the numbers below are a snapshot and not a final result. Table III gives selected epochs and Figure. 2 plots all of them.

Epoch	Precision	Recall	mAP50
1	0.008	0.268	0.034
5	0.470	0.249	0.108
10	0.648	0.249	0.161
15	0.614	0.256	0.203
18	0.622	0.286	0.235
21	0.577	0.255	0.238
23	0.624	0.276	0.244
24	0.563	0.267	0.231

Table 3: Validation Scores at Selected Epochs

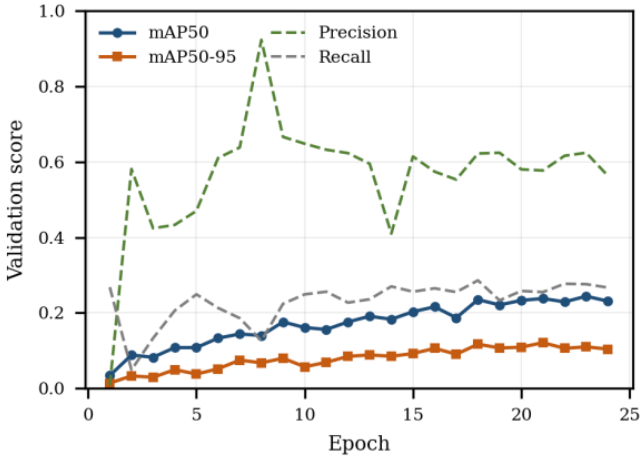


Figure 2: Validation Scores for Each Completed Epoch.

The highest mAP50 was 0.244 at epoch 23, where precision was 0.624 and recall was 0.276. The stricter mAP50-95 was highest at epoch 21 (0.121); by the library's default way of ranking epochs (mostly mAP50-95) this is the epoch whose weights would normally be saved as the best model. The training losses fell over the same period: the box loss went from 2.153 at epoch 11 to 1.995 at epoch 24, and the class loss from 2.289 to 1.826.

B. Detection and Counting Example

As a quick check, the trained model was run on one validation image with a confidence limit of 0.25. It found three plastic bottles and the whole step took 154.9 ms on the CPU, plus about 10 ms for reading and cleaning up the result. Counting the boxes therefore gave three bottles for that image. This is one example only and is not a measure of counting accuracy; a proper count test would compare predicted and true counts over many images, for example with mean absolute error.

C. Discussion

The precision of about 0.6 means that roughly six out of ten boxes the model draws are right, while the recall of about 0.27 means it finds only about one in four real objects. In plain terms, the model is careful but misses a lot, so its counts would tend to be too low. The mAP50 of 0.244 is well below what larger models reach on cleaner data sets, and this paper does not claim otherwise.

Several causes are likely. First, 371 training images is small for seven classes. Second, the smallest YOLO model was used, which trades accuracy for speed. Third, two of the seven labels look like placeholders and may confuse the training. Fourth, water scenes are hard: glare, ripples, reflections and partly sunk items change how garbage looks. Finally, training stopped early in the learning curve. Even so, the model is small, runs on an ordinary CPU, and learned useful patterns quickly, which supports using this approach for low-cost monitoring.

The study has clear limits. The result is a snapshot from an unfinished run, there are no per-class scores, the test folder was not used, only one example image was used to show counting, and no comparison with other detectors was made.

5. Conclusion

This paper built a simple pipeline that detects, names and counts floating garbage using a small YOLOv8 model. In the logged training run, the model reached an mAP50 of 0.244 (precision 0.624, recall 0.276) after 24 epochs on 371 training images, using only a CPU. The results are modest but show that the pipeline works end to end and is cheap to run.

Future work includes finishing the full training run and reporting results on the test folder, cleaning the class list, adding many more images from different weather and light conditions, trying larger YOLOv8 models and other detectors, reporting per-class scores, and checking counts against hand-counted images. Running the model on a drone or boat camera would be the final real-world test.

References

1. Jambeck, J. R., Geyer, R., Wilcox, C., Siegler, T. R., Perryman, M., Andrady, A., ... & Law, K. L. (2015). Plastic waste inputs from land into the ocean. *science, 347*(6223), 768-771.
2. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
3. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems, 28*.
4. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, September). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Cham: Springer International Publishing.
5. Fulton, M., Hong, J., Islam, M. J., & Sattar, J. (2019, May). Robotic detection of marine litter using deep visual detection models. In *2019 international conference on robotics and automation* (ICRA) (pp. 5752-5758). IEEE.

6. Proença, P. F., & Simões, P. (2020). Taco: Trash annotations in context for litter detection. *arXiv preprint arXiv:2003.06975*.
7. Cheng, Y., Zhu, J., Jiang, M., Fu, J., Pang, C., Wang, P., ... & Bengio, Y. (2021, October). Flow: A dataset and benchmark for floating waste detection in inland waters. In *2021 IEEE/CVF International Conference on Computer Vision* (ICCV) (pp. 10933-10942). IEEE.
8. Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics yolo. Zenodo.
9. Robo flow Universe, "Object-Detection, version 1," workspace small-intestine-cancer, CC BY 4.0.
10. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Cham: Springer International Publishing.

Copyright: *©2026 Prathamesh Garud, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*