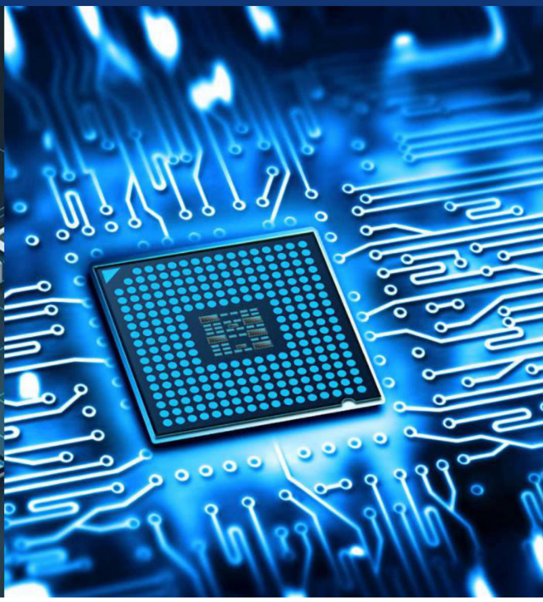
IJIRCCE

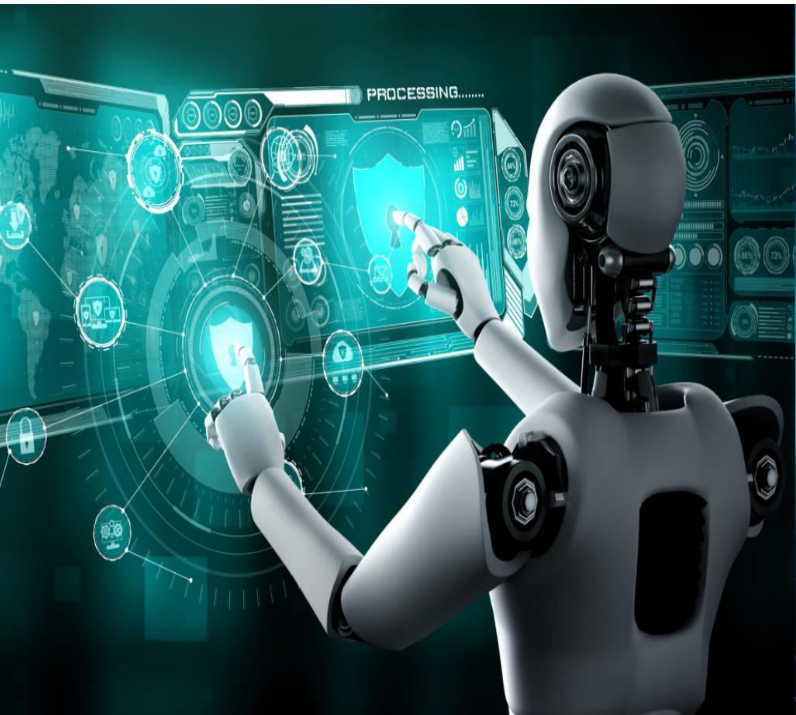
ISSN(O): 2320-9801

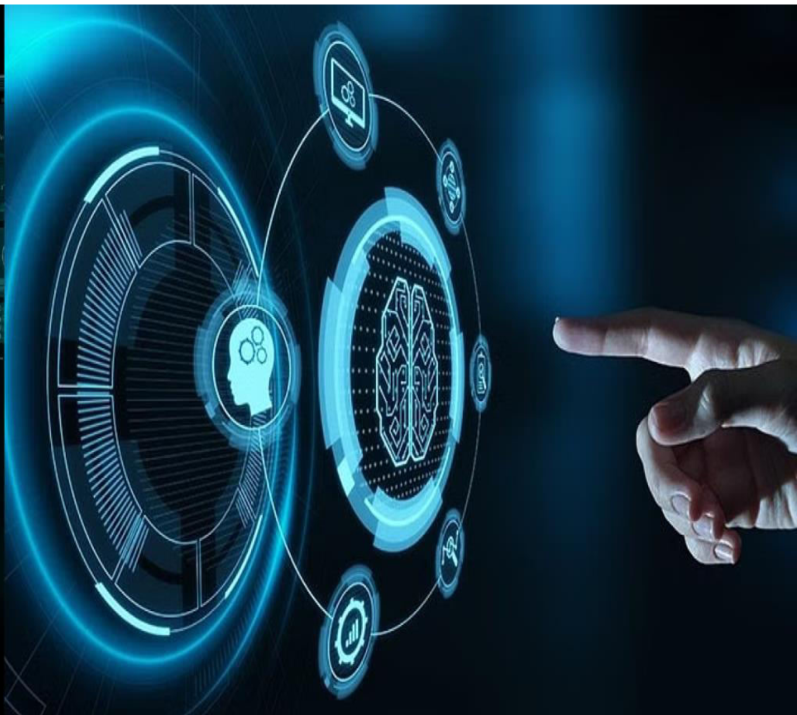
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771

Volume 14, Issue 3, March 2026

© 2026 IJIRCCE | Volume 14, Issue 3, March 2026 DOI: 10.15680/IJIRCCE.2026.1403068

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Implementation Paper on Detection of Plastic Debris in River Environments using Deep Learning

Prof. V.S.Sonawane, Bhargavi Bhagwat, Sanskruti Kothari, Harshala Gaikwad

Department of Artificial Intelligence & Machine Learning, AISSMS's Polytechnic, Pune, Maharashtra, India

ABSTRACT: Plastic pollution in river environments has emerged as a critical ecological concern, with rivers serving as primary pathways for transporting waste from land to oceans. Traditional monitoring methods are manual, time-consuming, and unable to provide timely assessments. This project presents RiverGuard AI, a deep learning-based system for automated detection of plastic debris in river imagery. The dataset, sourced from Roboflow, consists of annotated images of plastic waste under various environmental conditions. The YOLOv8n model was trained in Google Colab and integrated into a Python Flask web application with an HTML/CSS/JavaScript frontend. Users can upload river images through the browser interface and receive real-time detection results including annotated bounding boxes, plastic count, AI confidence score, and a severity-based remediation suggestion. The system demonstrated successful detection across varying pollution levels, classifying results as Clean, Mild, Moderate, or Critical with corresponding cleanup recommendations. The proposed system shows that deep learning, combined with structured datasets and lightweight web frameworks, can deliver an effective and accessible solution for automated river plastic monitoring.

I. INTRODUTION

Plastic pollution in rivers is a growing environmental crisis, with rivers acting as major pathways for transporting waste into oceans. Traditional monitoring methods are manual, slow, and limited in coverage, making timely response difficult.

With advancements in deep learning and computer vision, automated image-based detection has become a practical alternative. Convolutional neural networks, particularly YOLO- based architectures, have shown strong performance in real-time object detection tasks and are well suited for identifying plastic debris in river imagery.

This project develops RiverGuard AI, a web-based plastic detection system that uses a YOLOv8n model trained on a Roboflow dataset. Users upload river images through a browser interface built with Flask and HTML/CSS/JavaScript, and the system returns annotated results with a severity assessment and cleanup suggestion. The goal is to provide an accessible, automated tool that supports faster and more accurate environmental monitoring of river plastic pollution.

II. LITERATURE REVIEW

1. Overview

Several studies have explored the use of computer vision and deep learning for environmental monitoring, particularly for detecting plastic pollution in aquatic environments. This chapter reviews key works that influenced the design and methodology of this project.

2. Deep Learning for Object Detection

Redmon et al. introduced the YOLO (You Only Look Once) architecture, which revolutionized real-time object detection by processing images in a single forward pass through a convolutional neural network. Subsequent versions, including YOLOv5 and YOLOv8, improved detection speed and accuracy significantly. Jocher et al. (2023) released YOLOv8 through Ultralytics, offering a modular and easy-to-deploy framework suitable for custom detection tasks. These advancements made it feasible to apply real-time detection to environmental monitoring applications such as plastic debris identification.

3. Plastic Detection in Aquatic Environments

Majchrowska et al. (2022) developed a dataset and benchmark for floating garbage detection in water bodies, demonstrating that convolutional neural network-based models could achieve reliable detection even under challenging lighting and reflection conditions. Tharani et al. (2023) applied YOLOv5 to detect marine plastic waste from drone imagery, achieving high precision scores and validating the suitability of YOLO-based models for this domain. These

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

studies confirmed that annotated image datasets combined with modern detection architectures can effectively identify plastic debris in natural water environments.

4. Dataset Tools and Annotation Platforms

Dwyer et al. highlighted the role of platforms like Roboflow in simplifying dataset creation, annotation, augmentation, and export for computer vision projects. Roboflow supports direct export in YOLO-compatible formats, reducing preprocessing overhead and enabling faster model development cycles. Its use in this project allowed efficient dataset preparation without manual annotation scripting.

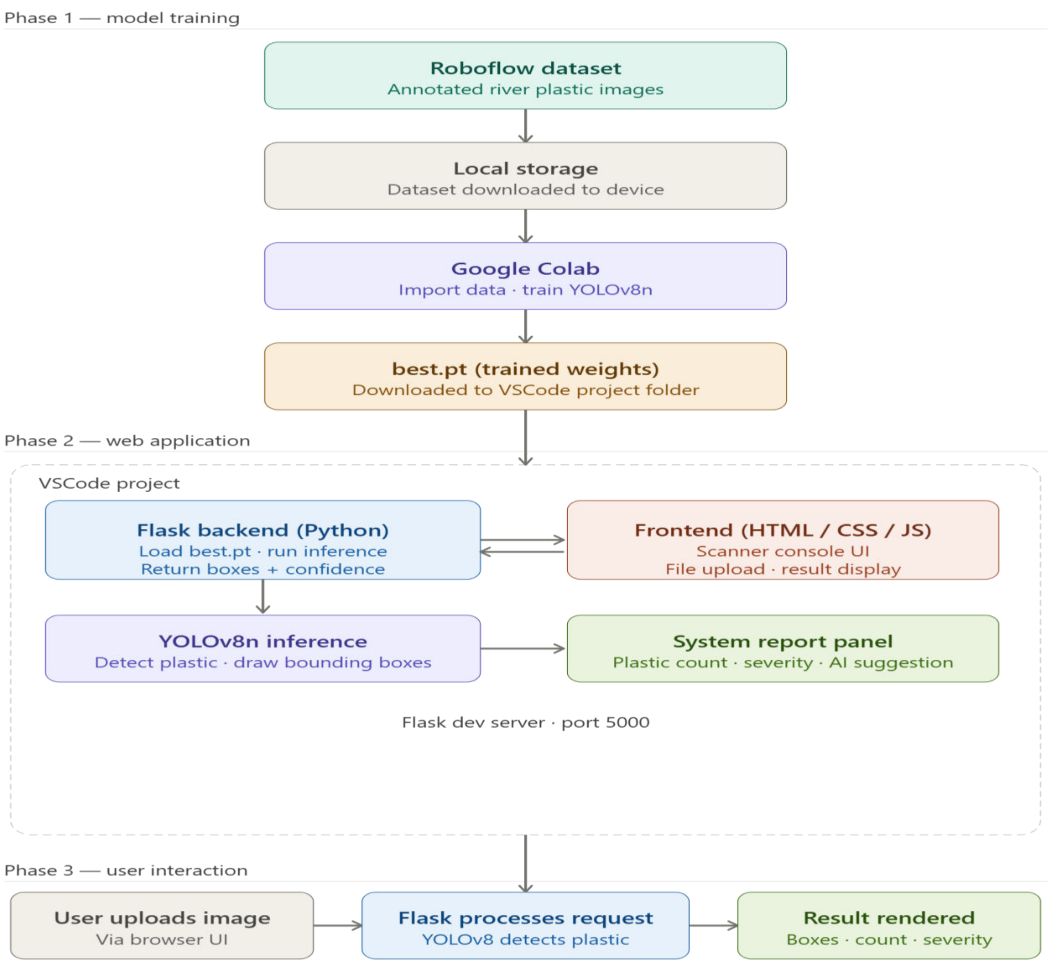
5. Web-Based Deployment of AI Models

Grinberg (2018) established Flask as a lightweight and flexible Python framework suitable for deploying machine learning models as web services. Subsequent works demonstrated that combining Flask backends with HTML/CSS/JavaScript frontends provides an effective approach for building accessible AI-powered applications without requiring complex infrastructure. This pattern was adopted in the present project to deliver detection results through a browser-based interface.

6. Summary

The reviewed literature confirms that deep learning, particularly YOLO-based architectures, is well suited for plastic debris detection in river and aquatic environments. Tools such as Roboflow streamline dataset preparation, while Flask-based web frameworks provide a practical deployment path. This project builds upon these established approaches, combining them into an integrated system tailored for real-time river plastic monitoring.
in river environments.

III. SYSTEM ARCHITECTURE



www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The architecture is split into three clear phases based on exactly how you built it:
Phase 1 — Model training: Roboflow → downloaded locally → imported into Google Colab → YOLOv8n trained → best.pt downloaded back to your machine.
Phase 2 — Web application (VSCode): Your project folder holds both the Flask backend and the HTML/CSS/JS frontend. Flask loads best.pt, handles the image POST request, runs YOLOv8 inference, and sends back the annotated image + detection data. The frontend scanner console displays results — plastic count, severity level (mild/moderate/critical), and the AI suggestion text.
Phase 3 — User interaction: User opens 127.0.0.1:5000 in Chrome, uploads a river image, Flask+YOLOv8 processes it, and the annotated result renders live in the browser — exactly what your screenshots show.

IV. IMPLIMENTATION

1. Overview

The implementation of the RiverGuard AI system was carried out in three phases: dataset preparation, model training, and web application development.

2. Dataset Preparation

The dataset was obtained from Roboflow, which provided annotated images of plastic debris in river environments. The dataset was downloaded to the local machine in YOLOv8- compatible format, containing train, validation, and test splits with corresponding YOLO annotation files. Preprocessing steps including resizing to 640×640 pixels, normalization, and augmentation were applied automatically during training.

3. Model Training

The dataset was imported into Google Colab, where the Ultralytics YOLOv8n model was trained using GPU resources. The model was trained for 50 epochs with a batch size of 16. After training, the best-performing weights were saved as best.pt and downloaded to the local machine.

4. Web Application Development

The project was set up in Visual Studio Code. The best.pt file was placed in the project folder alongside a Python Flask backend (app.py) and an HTML/CSS/JavaScript frontend. The Flask backend loads the trained model at startup, accepts image uploads via a POST request to the /predict endpoint, runs YOLOv8 inference, and returns the plastic count, confidence score, severity level, and annotated image. The frontend provides a Scanner Console interface where the user uploads a river image and clicks START AI SCAN. Results are displayed dynamically, showing the annotated image with bounding boxes, plastic count, AI confidence, and an AI suggestion based on severity — mild triggers a manual cleaning recommendation, moderate triggers a scheduled cleanup advisory, and critical triggers deployment of containment measures.

5. Deployment

The application was run locally using Flask's built-in development server on port 5000. Accessing http://127.0.0.1:5000 in a browser opens the RiverGuard AI interface, completing the end-to-end pipeline from image upload to real-time plastic detection and environmental assessment.

V. RESULTS

The result of the project shows that the deep learning models can be effective used for the environmental applications such as plastic detection in river detection. The system demonstrate how image processing and machine learning techniques can help monitor plastic in the river bodies and take the early preventive measure using the our model.

© 2026 IJIRCCE | Volume 14, Issue 3, March 2026 DOI: 10.15680/IJIRCCE.2026.1403068

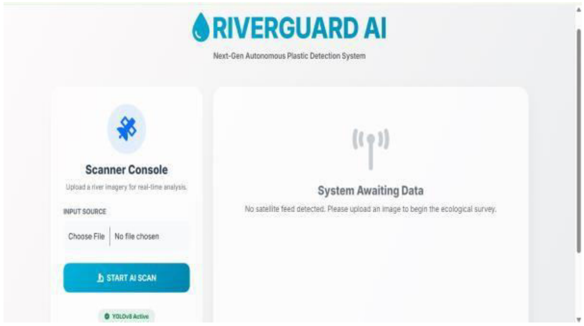
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

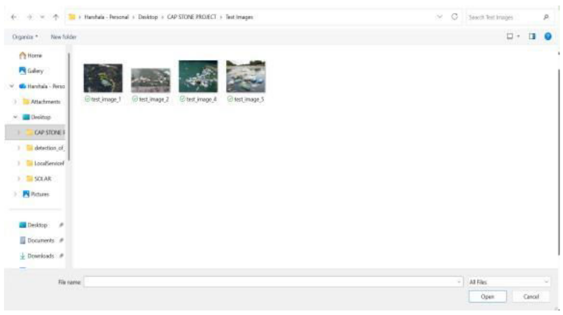
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Working

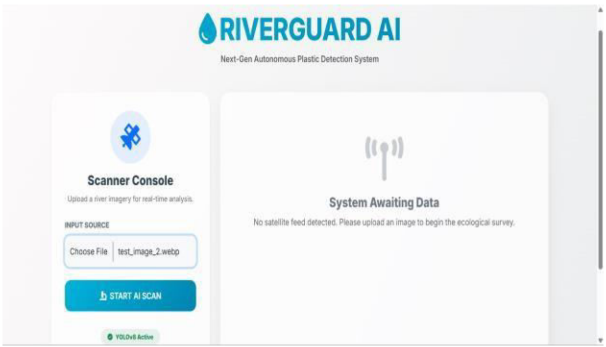
1) Frontend



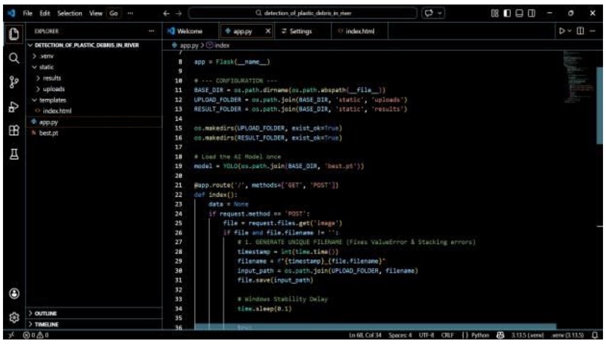
2) Uploading Image



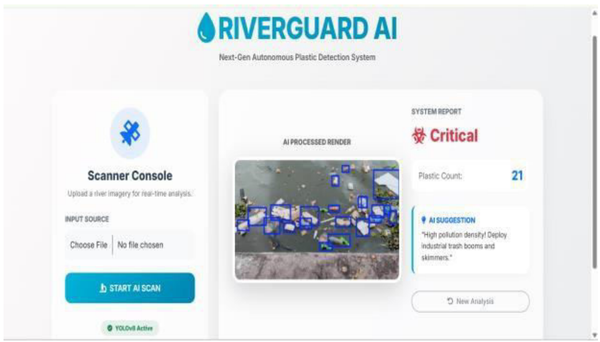
3) Starting Scan



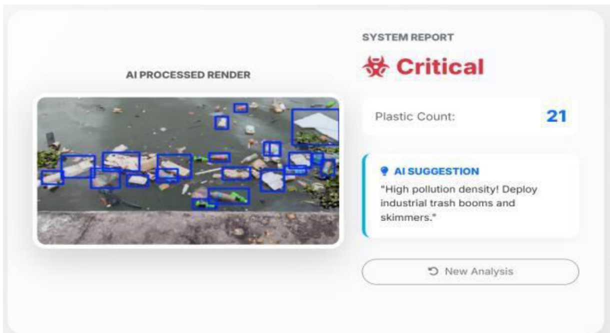
4) Processing



5) Output



6) System Report and AI Suggestion



VI. CONCLUSION

This project successfully designed and implemented RiverGuard AI, an automated deep learning- based system for detecting plastic debris in river environments. The system was built through a structured pipeline that began with dataset collection from Roboflow, followed by model training using the YOLOv8n architecture in Google Colab, and culminated in the development of a fully functional web application using Python Flask and HTML/CSS/JavaScript.

The trained model demonstrated the ability to accurately detect and localize plastic objects in river imagery by drawing bounding boxes around identified debris. The Flask backend efficiently handles image uploads, runs real-time inference using the trained weights, and returns detection results to the frontend. The browser-based interface provides users with

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

a clear and informative report including the total plastic count, AI confidence score, severity classification, and a specific cleanup recommendation tailored to the detected pollution level.

VII. FUTURE SCOPE

While the current system performs well for static image-based detection, several enhancements can be explored in future work:

- Live video detection — Extending the system to process real-time video streams from CCTV cameras or drones mounted over rivers.
- Mobile application — Developing an Android or iOS app to allow field workers to capture and analyze river images directly from smartphones.
- GPS-based pollution mapping — Integrating geolocation data with detection results to generate interactive pollution heatmaps of river networks.
- Improved model accuracy — Training on a larger and more diverse dataset to improve detection under challenging conditions such as water reflections, low lighting, and partially submerged plastic.

REFERENCES

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*
2. Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8. Retrieved from https://github.com/ultralytics/ultralytics
3. Majchrowska, S., Mikołajczyk, A., Ferlin, M., Klawikowska, Z., Plantykow, M., Kwasigroch, A., & Majek, K. (2022). Deep learning-based waste detection in natural and urban environments. *Waste Management, 138*, 274–284.
4. Tharani, S., Maheshwari, A., & Sharma, P. (2023). Detection of Marine Plastic Waste Using YOLOv5 and Drone Imagery. *International Journal of Environmental Science and Technology.*
5. Dwyer, J., & Roboflow Team. (2023). Roboflow: A Platform for Computer Vision Dataset Management. Retrieved from https://roboflow.com
6. Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python* (2nd ed.). O'Reilly Media.
7. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature, 521*(7553), 436–4.