Develop an Automated system for River Water Pollutant Detection and Classification to Alert the Authority using a Deep Learning Framework

[*1]Neeta Shirsat, [2]V. Nirmalrani

[*1]Research Scholar, Department of Computer Science and Engineering, Sathyabama Institute of Sci and Tech, Chennai, India.

[2]Professor, Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

Email-ID: [1]neetaneeta09@hotmail.com , [1*]neeta.shirsat@gmail.com

[2]nirmalrani.it@sathyabama.ac.in

Abstract

Pollution has become an increasingly critical environmental issue, with direct, severe impacts on human health. It increases due to modern lifestyles, the use of non-disposable items, increased industrialization and urbanization, and poor waste management. Traditional pollution monitoring systems are time-consuming, labour-intensive, lack scalability, and are difficult to use for real-time early identification. The primary objective of this paper is to design and implement an automated waste monitoring system for detecting and classifying various pollutants present on the river surface using a deep learning framework. A custom dataset of river images was created using multiple sensor-based IoT devices, such as drones and CCTV cameras, to identify and detect different pollutants, including wood, paper, plastic debris, use-and-through items, oil films, foam, and other industrial waste. This paper trains the Fast Region-based Convolutional Neural Network model using the dataset to perform object detection and classification in parallel. It also integrates with a real-time alert system that notifies authorized personnel of pollutants detected in a specific region. The Fast-RCNN demonstrated high detection and classification accuracy, with very few false positives, on test and validation datasets derived from real-world river images. The input data is collected from the publicly available Kaggle river pollution detection dataset, which includes 70 raw images after being augmented, expanded to 6000 images with a uniform pixel size of 224 × 224. The automated alert mechanism successfully notified authorities of the precise location and type of pollutants. This study presents a scalable, real-time, and automated river pollutant-detection system that combines DL with environmental surveillance technologies. Future work will focus on integrating chemical sensor data and deploying the system in large-scale smart city environments.

Keywords: Waste Management, Solid Waste Classification, Waste Hazards in Floating Water, River Water Cleaning.

1. Introduction

Managing waste in the river is crucial to ensuring the sustainability of the ecosystem and the communities that depend on it. Numerous types of solid and liquid waste contaminate rivers through human actions and natural phenomena (Denchak (2018)). A mixture of direct and indirect waste dumping and water discharge causes these wastes to be deposited in floating water. These water pollutants arise from either point or diffuse sources (Cory Ochs et al. (2024)). Point sources are identifiable, such as discharge pipes from industries, and dispersed sources are expansive, uncontained areas where pollutants enter the water body, such as water runoff from agricultural fields (Dwivedi (2017)). One primary type of pollutant is macro-plastics, which are larger than 5mm. Roughly eight tonnes of plastic waste are likely to enter the land and rivers in Prayagraj, approximately 500 kilometres from Agra (Maharjan et al. (2022)). These smaller particles settle in the river and are ingested by aquatic organisms, such as fish, damaging their internal organs, inducing oxidative stress, and impairing their immune system and reproductive capabilities. Researchers found extensive microplastic contamination in the Ganges River, with around 41 microplastics per square metre daily. Sediment samples contained an average of 57 microplastics per kilogram, and one microplastic particle per 20 Liters (Siegfried et al. (2017)).

Figure-1. Waste Hazards in Floating Water

This type of water pollution can reduce the GDP growth of downstream regions by one-third (Khan et al. (2021) and (Das, A. (2025)). Riverbank erosion, habitat destruction, and sedimentation also pose water hazards that disrupt rivers and their ecosystems (Das, A (2024)). All these individual problems can strain the river's health, and combined, they make it challenging to manage ecological and human systems at their boundaries. National River Revitalization Programs aim to restore and protect river ecosystems by improving water quality, biodiversity, and overall river health (Gani et al. (2025)). Modern technologies like remote sensing, data analysis, and GIS mapping enhance water quality monitoring, pollutant tracking, and flood prediction, making river management more efficient (Chatrabhuj et al. (2024)). Pollution along the riverside, as shown in Figure-1, includes various types of pollutants. Although the current state of DL-based waste detection is highly advanced, still it has multiple shortcomings such as limited range of datasets, lack of scalability in real-time, and high false-positive rates, as well as the basic need to detect and not to classify waste and provide automated response mechanisms. Most methods are only effective when applied to controlled databases or small-scale databases and not when used on dynamic rivers that include different lighting conditions, reflections and background complexity. To solve these issues, the current research will suggest a hybrid DL architecture using Faster R-CNN with CNN as an accurate method to detect and classify hazardous and non-hazardous floating waste.

Research Objective

The objectives of the paper are to develop an automatic alert system for waste management in smart cities. Use IoT devices for surveillance and capture images and videos in floating water environments, such as rivers. Design and implement a robust deep learning (DL) model for analyzing and predicting waste materials from the river images. The performance of the proposed (FRCNN) model is compared with the output of other neural network models, such as CNN, RPN, and RCNN.

Following the introduction and Objective above, this paper is organized as follows. Section 2 presents the Background study and Research Gap; Section 3 explains the proposed FRCNN model in detail. Section 4 presents the experimental results and compares performance with the study's limitations. Finally, section 5 concludes that the proposed work is robustly practical.

2. Background Study

The Global Peace Foundation (GPF) India launched a cleanup initiative at the Yamuna River in Wazirabad on February 23, 2025 (Ravish et al. (2025)). River water quality monitoring assesses its suitability for various uses by evaluating physical, chemical, and biological parameters, including oxygen, pH, turbidity, temperature, nutrient levels, and pollutants (Mohamad Ali Fulazzaky, 2010; Patil P. N et al., 2012). Most studies consistently show that contaminant levels are continuously increasing from household, agricultural, and industrial processes. The authors in Ogidi & Akpan (2022) and Sonone et al. (2020) explain that waste from heavy metals and non-disposable pollutants has a greater negative impact on environmental cleanliness and human health. These limitations motivate the design and implementation of broader, scalable, automated surveillance and analytical models based on robust deep learning algorithms for waste detection and classification. Ayhan Demirbas (2011) investigated several waste management methods, highlighting that waste reduction, recycling, and reuse are the most essential for improving floating water environments. Mostaghimi & Behnamian (2023) and Pihlajarinne (2021) have provided only conceptual frameworks with limited practical evidence. Parkinson & Thompson (2003) focused on recycling waste materials, and Khalid et al. (2022) explained waste disposal methods. It shows that the traditional methods used for waste management are insufficient and inefficient, and need to be integrated with advanced intelligent methods and automation to provide a better solution for the current study.

H.J. parkinson and G. Thompson (2003) investigated the recycling process. A limitation of their work is that it focuses on theory rather than providing real-time insights. Ibrahim Khalid et al. (2022) reviewed the overall concept of disposal. They incorporated sustainable methods to reduce environmental and public health impacts. The limitation of this research is its narrow scope of

industries covered, which requires considerable time for the study, and its reliance on specific remanufacturing definitions. (Moustafa A. Chaaban, 2001). An effective approach involves addressing waste at its source. Agricultural runoff, which contains excessive amounts of fertilizers and pesticides, is a significant contributor to river pollution. Advances in biotechnology highlight the potential of science-based techniques for large-scale environmental restoration.

Recently, ML and DL algorithms have shown promising results in detecting and classifying surface waste hazards in floating water bodies. For example, McShane et al. (2021) and Venkatesan & Krishnan (2025) have implemented a TensorFlow model integrated with CNN for classification, Sio et al. (2022) have used YOLO-v5 for waste prediction and classification, Lieshout et al. (2020) and Faisal et al. (2022) developed deep learning frameworks for automatic waste detection and classification with high accuracy and demonstrated more progress. However, they faced various challenges, including limited datasets, environmental variability, inadequate real-time adaptability, and inconsistent prediction accuracy. These limitations motivate this work to improve detection and classification accuracy through optimization and a deep learning algorithm, making the model highly suitable for floating water environments. Recently, Armitage et al. (2022) proposed an advanced video-based method for detecting microplastic waste; Yang et al. (2024) implemented an improved CNN model for classifying waste types; and Chellaiah et al. (2024) implemented a hybrid CNN-LSTM model for pollutant monitoring. Some of the authors in (Thavasimuthu et al., 2024; Arepalli & Naik, 2024; BENDIB & Dhia, 2024; Hou et al., 2024) have used advanced neural network and YOLO architectures, such as SegNet-VOLO, DST-CNN, YOLO-v8, Mask-RCNN, and ENS-YOLO-v8 for increasing the prediction accuracy for all kinds of pollutants from the input data. These advanced architectures have provided high accuracy only for certain types of contaminants, not for all. They performed well only on small, specific, and restricted datasets; thus, they are not robust in real-world experimental validation. The performance comparison of the earlier works is given in Table-1.

Table-1. Performance Summary

Author and Year	Research Method	Accuracy (%)
McShane et al. (2021), Venkatesan & Krishnan (2025)	TensorFlow and a CNN for classifying floating waste in a river.	92.30%
Faisal et al. (2022)	Fast R-CNN for detecting floating plastic waste	90.50%
Gilroy Aldric Sio et al. (2022)	YOLOv5 for detecting and classifying garbage and plastics in floating water	84.30%
Armitage et al. (2022)	CNN-based microplastic detection.	95.20
Yang et al. (2024)	CNN and VGG-16 for floating object detection and classification.	93.86
Chellaswamy Chellaiah et al. (2024)	CNN-LSTM for pollution detection in real-time river water.	98.4
Rajendran Thavasimuthu et al. (2024)	SegNet-Vision Outlooker (VOLO) model.	98.2
Peda Gopi Arepalli and K. Jairam Naik (2024)	Dilated Spatial-temporal CNN for waste detection in a real-time dataset.	99.28
BENDIB and Mohamed Dhia (2024)	YOLOv8 and Mask RCNN for underwater plastic waste detection.	84%
Qingzhi Hou et al. (2024)	Enhanced Network Structure with YOLO-v8 for Waste Detection	92.5%.

Research Gap

Particularly in developing nations, where untreated industrial waste and household garbage are often dumped into water bodies, river pollution has become a serious environmental and public health concern. Timely interventions are delayed by the laborious and non-scalable. However, most earlier studies have focused solely on detecting waste hazards, rather than categorizing them. This paper aims to detect and classify waste hazards by addressing these aspects, thereby enhancing the overall effectiveness of the proposed work. To analyse photos and videos captured by drones and fixed cameras, this study proposes a deep learning-based automated framework for river pollution detection and classification that leverages Convolutional Neural Networks (CNNs) and Faster R-CNN (FRCNN). Using communication protocols such as SMS, email, or dashboard notifications, the objective is not only to detect various types of pollutants but also to issue real-time alerts to authorities. To transform river water quality monitoring and aid in the creation of intelligent, sustainable environmental management systems, the proposed method addresses the challenges of precise categorization, ecological variability, and automated alarm systems.

3. Proposed Methodology

This paper presents an FRCNN model for dynamic pattern recognition and image detection, aiming to classify normal and hazardous waste in input floating water images. The overall workflow of this paper is illustrated in Figure-2. The input video frames are initially segmented into multiple frames to train the model. This helps the model accurately detect and classify hazardous (plastic) and non-hazardous waste on the surface of the floating water. The segmented image is then pre-processed to enhance image quality. Objects are detected and boxed from the pre-processed data using the FRCNN model. Finally, the detected bounding boxes are analysed using the CNN model and classified into hazardous (plastics) and non-hazardous (other materials).

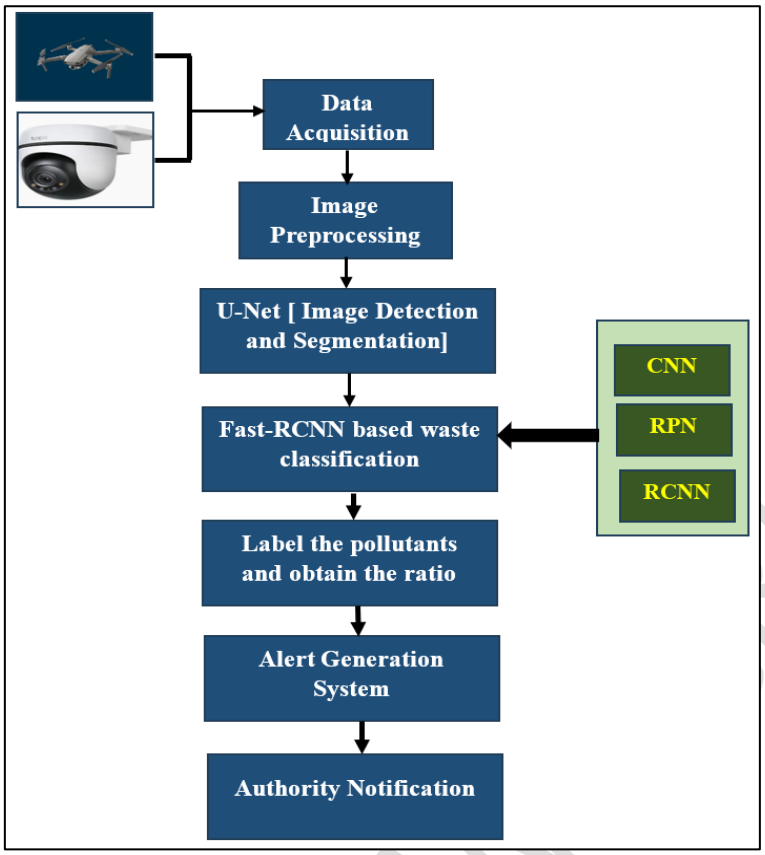


Figure-2. Proposed Workflow

3.1 Video to Image Segmentation

It is not possible to process a video directly. Thus, the input video file is converted into a sequence of video frames, also known as images. If a video plays at f frames per second (fps) and runs for T seconds, the total number of frames N can be calculated as **N = f × T**, where f is the frame rate. Key frame is selected from every k^{th} frame, where k is a desired interval. This gives you approximately N/k images. For example, taking every 5th frame from a video running at 30 frames per second (fps) will result in images captured at 6 frames per second (fps). The frames are applied for further image processing. To understand video-to-image conversion, refer to Figure 3, which illustrates how a 1-second video is converted into a sequence of video frames.

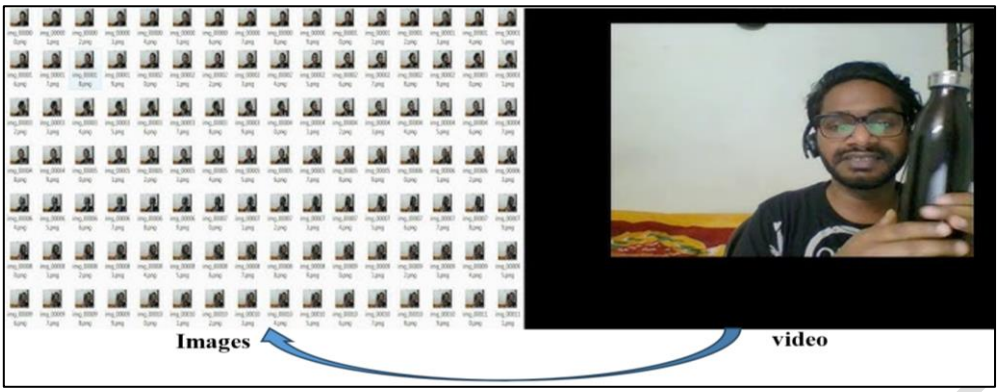


Figure-3. Video To Image Conversion

3.2 Data Pre-Processing

In a DL-based model, data preprocessing is a crucial step in enhancing the quality of the input image. The input floating image data is collected from various regions, angles, shapes, and sizes. The data preprocessing step eliminates these artifacts by applying image cropping, resizing, denoising, and contrast enhancement.

3.2.1 Image Cropping

Cropping reduces a system's computational requirement and increases object detection accuracy.

$$I_{crop} = I[y_1 : y_2, x_1 : x_2, :]$$

In the above equation, $(x_2, y_2)\ and\ (x_1, y_1)$ represents the bottom-right and top-left coordinates of the ROI.

3.2.2 Image Resizing

Resizing sets the input image dimensions to a fixed value, which is essential for feeding DL models.

$$I_{resize}(x'y') = I(\frac{x'}{W'} \cdot W, \frac{y'}{H'} \cdot H)$$

Bicubic and bilinear interpolation are the most common interpolation techniques.

3.2.3 Image Denoising

Noise in uncontrolled environments can appear in images due to low light, water splashes, or limitations of the camera sensor.

$$I_{denoise}(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{j} G(i,j) \cdot I(x+i, y+j)$$

In the above equation, $G(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$ is the Gaussian kernel. NLM (Non-Local Means) can be an alternative if the noise characteristics are known.

3.2.4 Contrast Enhancement

Contrast improvement helps make floating waste objects stand out against water backgrounds, particularly when contrast is minimal. One popular method was histogram equalization, which distributes pixel intensities.

$$P_{eq}(i) = [(L-1) \sum_{j=0}^{i} \frac{n_j}{N}]$$

In the above equation, n_j represents the number of pixels with the intensity j, L represents the number of grey levels, and N represents the total number of pixels.

3.3. Data Augmentation

Synthetic variations of training images are formed to enhance model generalization and robustness.

$$I_{aug} = T(I)$$

Were, $T \in \{R_\theta, S_{(sx,sy)}, F, B_\delta, N\}$ denoted the rotation obtained by θ brightness, flipping is shifted by δ and scaling by (sx, sy) or sometimes added noise by using δ. The rotation is evaluated using the following equation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

By applying the above steps, the input image quality improves. It is also verified in various scenarios of real-time applications.

3.4. Waste Object Detection in Floating Water

Deep learning models, such as U-Net, are widely used to detect floating waste objects because they excel at image segmentation. This task involves identifying objects such as plastic and debris on the water surface, and a smart system is needed to separate the waste from the water background.

3.4.1 Understanding U-Net

U-Net is a convolutional neural network (CNN) originally developed to analyze medical images. It uses an encoder-decoder structure with skip connections. These skip connections help the model retain important details while learning, which is useful for pixel-level segmentation. U-Net comprises two main parts: the encoder and the decoder. The encoder progressively reduces the image size while learning key features. The decoder increases the image size again to match the original and helps the model predict each pixel's label. The structure of the U-Net model is shown in Figure-4. The main goal of the U-Net in segmentation tasks can be written as:

$$Y = f(X; \theta)$$

Where X is the input image (in this case, an image showing waste floating on water), Y is the segmented output image (highlighting where the waste is), f(X; θ) is the function the model learns, and θ stands for the model's parameters or weights.

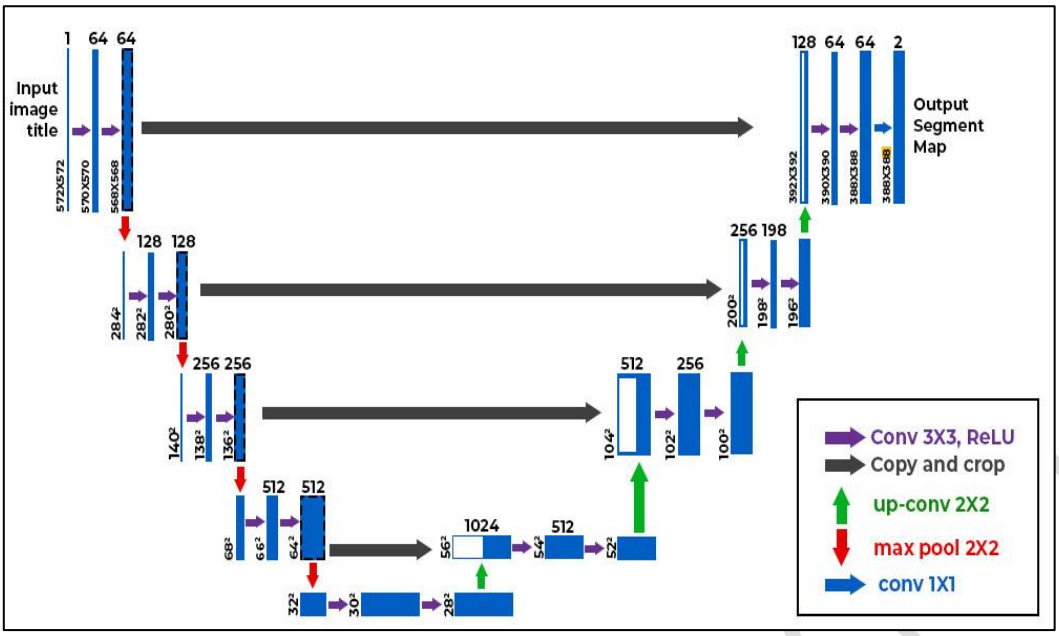


Figure-4. U-Net architecture

3.4.1.1 Encoder Stage

The encoder uses convolutional layers to extract features at different levels. Each layer in this stage can be described by:

$$C_l = \sigma(W_l * C_{l-1} + b_l)$$

Where, C_l is the feature map at the i-th layer, W_l represents the filter weights used for convolution, C_{l-1} is the feature map from the previous layer, b_l is the bias term added during the calculation. The bottleneck in a U-Net model is the part where the most detailed and abstract features are learned. It can be written as:

$$B = \sigma(W_b * C_{last} + b_b)$$

In the decoder, the model increases the size of the feature maps to match the input image. This process is shown as:

$$D_l = \sigma(W_l' * UP(C_l) + b_l')$$

Here, UP means upsampling (like transposed convolution), W_l' are the weights for the upsampling layers, and D_l is the result at each decoding step.

The last layer in the model is usually a 1x1 convolution. It reduces the number of output channels to match the number of classes (for example, 2 classes in binary classification: waste and background). This can be written as:

$$Y = W_f * D_{last} + b_f$$

Where W_f and b_f are the weights and biases for this layer.

3.4.1.2 Binary Cross-Entropy Loss:

$$\boldsymbol{L = -\sum_{i=1}^{N} [y_i log(\hat{y}_i) + (1 - y_i) log\ (1 - \hat{y}_i)]}$$

Where, y_i denotes the actual label (1 = waste, 0 = water), $\hat{y}_i$ denotes the model's predicted probability, and N is the total number of pixels.

Dice Loss:

$$\boldsymbol{D = \frac{2|A \cap B|}{|A| + |B|}}$$

Where A denotes the actual waste pixels, and B indicates the predicted waste pixels. This measures how well the predicted waste area overlaps with the true waste area. Once trained, the U-Net model can take a new image and produce a mask marking each pixel as waste or water. U-Net is designed to accurately label every pixel, which is essential for detecting waste in cluttered environments.

3.4.2 Faster R-CNN

Regional Proposal Network (RPN) - The RPN takes the input images and creates effective bounding boxes (proposals). Let the input image be I, and let the RPN task be to generate bounding box proposals. $\{B = b_1, b_2, \ldots, b_k\}$, where each $b_i = (x, y, w, h)$ shows the ensemble of bounding boxes, and (x, y) is considered to be the center of the box and w, h denotes width and height. RPN uses a sliding-window technique over the feature map at each position, and it evaluates several anchor

boxes (i.e., predefined bounding boxes with different aspect ratios and scales) that match ground truth objects. RPN predicts two things for each anchor box: they are

(a). Objectness Score P_i, that denotes the probability that the anchor box holds an object:

$$p_i = \sigma(W_{cls} \cdot f(I) + b_{cls})$$

Where σ is the sigmoid activation function, $W_{cls}$and b_{cls} denoting learned weights and biases for the classification layer and $f(I)$ is the image feature map generated by the CNN backbone (e.g., VGG, ResNet).

(b). Bounding Box Regression $\Delta b_i = (\Delta x, \Delta y, \Delta w, \Delta h)$, that is adaptable to the anchor box with high precision, matching the object in the image.

$$\Delta b_i = W_{reg} \cdot f(I) + b_{reg}$$

Where W_{reg} and b_{reg} denotes learned weights and biases for the bounding box regression layer and Δbi indicates the offset used in the anchor box to achieve greater bounding-box accuracy.

(c). RoI Pooling Layer - Once the bounding box proposals are generated by applying RPN, Faster R-CNN uses a RoI (Region of Interest) pooling layer to leverage a fixed-size feature map from these proposals.

$$Rol\ Pooling(R_i) = Resize(R_i, H, W)$$

Where, the i^{th} region of interest (bounding box proposal) is considered to be $R_i$and height and weight of the pooled output (formula) are taken as H, W. The structure of FRCNN is shown in Figure-5.

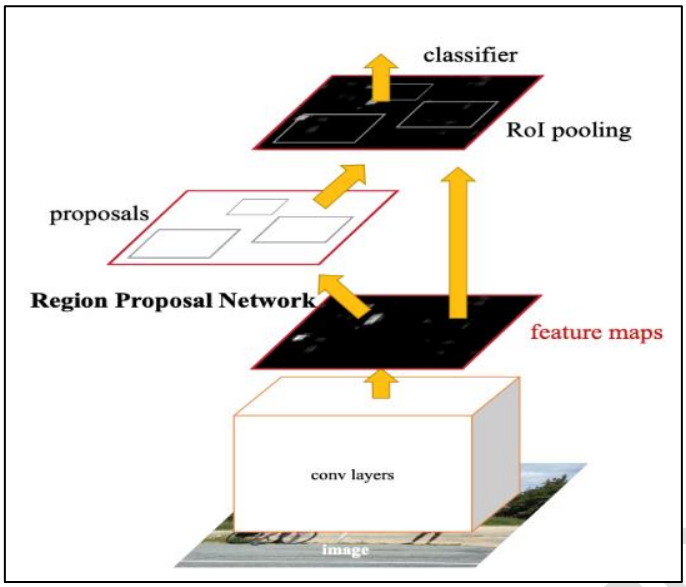


Figure-5 FRCNN architecture

3.4.2.1 Classification and Bounding Box Refinement

Every pooled region is passed through a fully connected (fc) layer, which produces the final classification and enhanced bounding box coordinates. R_i is the given region, the process here is expressed as,

Classification: For each class C (including the background class), the FC layer creates a score:

$$P(C|R_i) = Softmax(W_{cls} \cdot Rol\ Pooling(R_i) + b_{cls})$$

Where $P(C|R_i)$ denotes the probability of a class C for the R_i, as region and SoftMax ensure the output is a valid possible allocation across all classes.

Bounding Box Regression: The FC layer forecasts the adaptations to the bounding box coordinates:

$$\Delta b_i = W_{reg} \cdot Rol\ Pooling(R_i) + b_{reg}$$

These adaptations are applied to clarify the assumed bounding box, which results in a final bounding box, $B_f = (x_f, y_f . w_f, h_f)$

3.4.2.3 Faster R-CNN (FRCNN) for Object Detection Using Bounding Boxes

FRCNN (Faster R-CNN) is widely used in object detection architectures and is also known to be the most effective. It is built on the previous R-CNN (Region-based CNN) and FR-CNN models. However, FR-CNN introduced the RPN (Region Proposal Network) to generate region proposals, thereby speeding up the process and making it more practical for real-time object detection. The internal components of FRCNN are the RPN and ODN, which accurately detect objects in the input images.

3.4.3 CNN-Based Object Classification

The most widely used Deep Learning architecture is the Convolutional Neural Network (CNN), which is used explicitly for image classification. To categorize floating objects in river water, such as plastic bottles, wood logs, and clothes, a CNN processes image data to learn features such as shape, color, texture, and spatial relationships. The structure of the CNN model is shown in Figure 6.

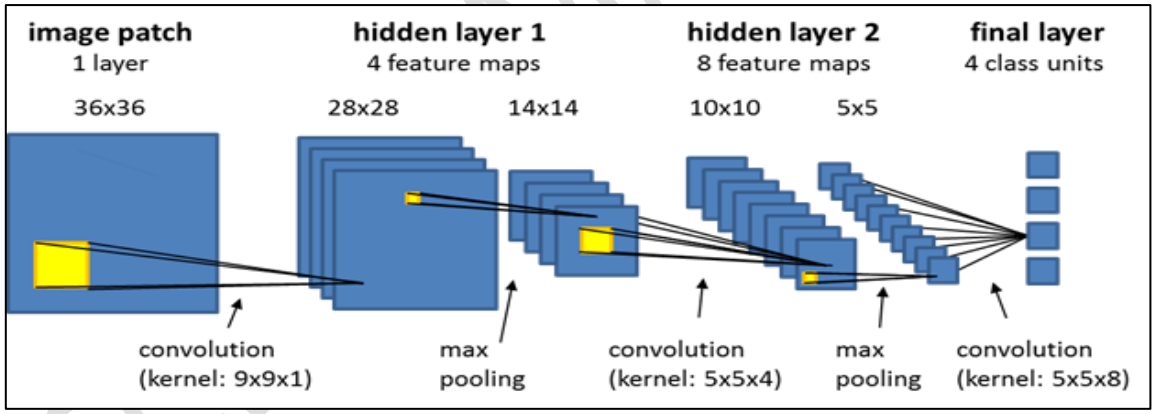


Figure-6: CNN model architecture

CNN classifies these floating objects using multiple layers, each with a separate function. The input of the CNN image in the format of $X \in R^{H \times W \times C}$, where height and width are denoted as H *and* W, the number of channels like RGB is denoted as C. For example, the dimensions of the input river image surface are $256 \times 256 \times 3$. The low-level features are detected by applying a set of learnable filters $k_i^{(1)} \in R^{K \times K \times C}$ in the initial layer. The output feature map $F_i^{(1)}$ is calculated for each filter i:

$$F_i^{(1)} = \sigma\left(X * k_i^{(1)} + b_i^{(1)}\right)$$

Where the convolutional operator is denoted as *, the bias term is denoted as $b_i^{(1)}$ and a non-linear activation function is denoted as σ, generally $ReLU$: $\sigma(x) = max(0, x)$.

$$P_i^{(1)} = MaxPool\left(F_i^{(1)}\right)$$

The network primarily focused on presence and location features rather than exact positions, using downsampling to improve the model's reliability for object translation and scale.

$$F_j^{(l)} = \sigma\left(\sum_i P_i^{(l-1)} * K_{ij}^{(l)} * b_j^{(l)}\right)$$

The network can learn higher-level representations of objects, such as floating plastic water bottles and leaf clusters. After the multiple convolutional layers, the 3D feature map is converted into a 1D vector $Z \in R^d$. This vector is transferred through one or more fully connected (dense) layers:

$$h^{(1)} = \sigma\left(W^{(1)}z + b^{(1)}\right)$$

$$h^{(2)} = \sigma\left(W^{(2)}h^{(1)} + b^{(2)}\right)$$

Where, the weight matrices are denoted as $W^{(1)}, W^{(2)}$ and bias are denoted as $b^{(1)}, b^{(2)}$. These layers are integrated for the object types and the corresponding configurations. Finally, a SoftMax layer is used for classifying over N object classes produces the probability distributions:

$$\hat{y}_i = \frac{e^{h_i^{(L)}}}{\sum_{j=1}^{N} e^{h_j^{(L)}}}$$

Where, for the class i the output of the dense layer is denoted as $h_i^{(L)}$, the predicted probability of the image for the class i is denoted as $\hat{y}_i$.

3.5 Multiple object detection

To combine object detection techniques such as YOLO, Region Proposal Network, or Fast R-CNN to classify multiple objects in a single image. The images are partitioned into regions like X_r. Each layer is then covered by the CNN layers discussed above, allowing the model to output both the object class and the bounding box coordinates. x, y, w, h forever detected a floating object. The overall CNN model functionality for detecting the floating objects is expressed as:

$$\hat{y} = f_{CNN}(X; \Theta)$$

Where the output probability vector is denoted as $\hat{y} \in R^N$, the entire weight and biases in the network are denoted as Θ. The minimizing loss function (cross entropy) is used to train the network:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

In terms of Θ, y is a true label vector.

3.6 Alert Generation Module

In this research work, generating an alert to the authorities is a significant component. Thus, the proposed model automatically alerts when hazardous waste is detected in the river water. Using surveillance footage from multiple regions, the proposed model detects dangerous and non-hazardous waste on the water's surface and generates alerts. The proposed model acts as a communication bridge between stakeholders and the detection engine for decision-making. This alert-generation process involves detecting pollutants in floating water and extracting pollutant information from input surface-water images. Hazardous waste on the water surface is detected based on the predicted object type, size, timestamp, and confidence score. Once hazardous waste is detected, the details are organized in a readable format and sent to the authority via email, SMS, or web dashboards. The overall working process of the proposed alert generation module is shown in Figure-8.

3.7 Performance Evaluation

We use standard performance measures, as explained in this section, to evaluate how well the detection models perform.

Mean Average Precision (mAP):

It's not always possible for predicted bounding boxes to perfectly match the actual (ground truth) ones because of differences in labeling and minor errors. So, we calculate a value called Average Precision (AP), which is the area under the curve when plotting precision against recall for each class. When this is averaged across all classes and different IoU levels, we get mAP. mAP helps evaluate how well an object detection model performs by measuring the overlap between the predicted and ground truth boxes. This overlap is called IoU (Intersection over Union), calculated as:

$$\boldsymbol{IOU = \frac{Area\ of\ Overlap}{Area\ of\ Union}}$$

The overlap area is where the predicted box and the actual box intersect. The union is the total area covered by both boxes combined. IoU values range from 0 to 1 — higher values indicate better matches. If IoU exceeds 0.5, the prediction is considered a correct detection (true positive or TP). If it is below 0.5 for all actual boxes, it is regarded as a false positive (FP). IoU remains effective even when the dataset is unbalanced. In this work, we use a 0.5 IoU threshold. mAP is widely used and serves as the primary metric for comparing models on datasets like COCO. AP indicates how accurate the predicted boxes are (precision) and how many real objects are detected (recall).

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

Precision refers to the accuracy of how many of the predicted objects are actually correct.

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

Recall refers to the percentage of actual objects that were successfully detected. FN (false negative) represents the model missed a real object (i.e., the predicted boxes did not overlap enough with the actual box).

Perfect detection would mean precision is 1 for every recall level. Typically, when you attempt to increase recall, precision decreases; conversely, when you increase precision, recall may drop. AP calculates the average precision over different recall values.

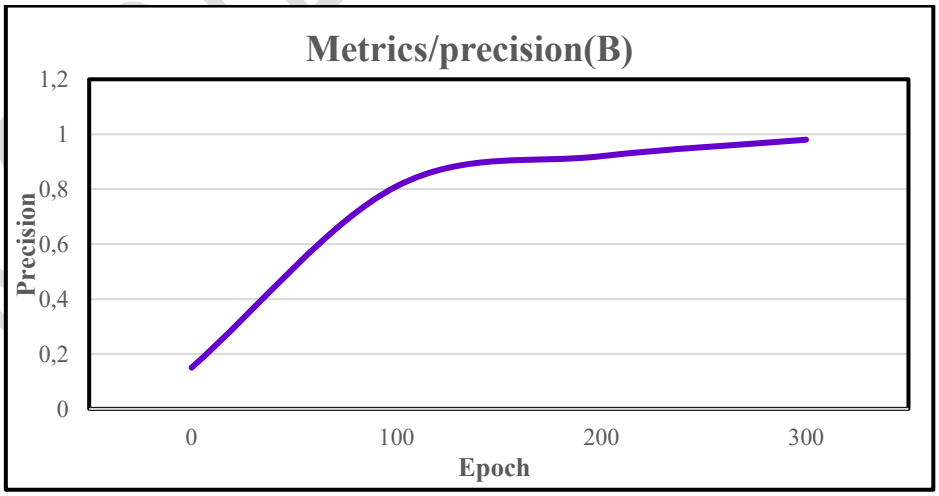
4. Experimental Results and Discussion

In this section, various simulation results for the proposed model are presented using the simulation software installed on a system equipped with a 1TB HDD, 64GB RAM, Windows OS, and an Intel i7 processor. The overall input data is collected from a publicly available "River Plastic Pollution Detection" Kaggle dataset [23]. The dataset includes 70 raw images collected using various sensors and devices. Further, data augmentation techniques, such as image flipping, rotation, contrast adjustment, and scaling, are applied to improve the model's detection efficiency. After data augmentation, all the input images are resized to a uniform size of 224×224 pixel. The augmented images have now been expanded to 6000 samples. To reduce model complexity, the input samples are split into three phases: training, testing, and validation, with the ratio of (70:15:15), that is, from the augmented image samples of 6000 images, 4200 images are used for training, 900 images are used for validation, and the remaining 900 is used for testing. This setup enables the development of AI-based tools to track and identify plastic waste in aquatic environments.

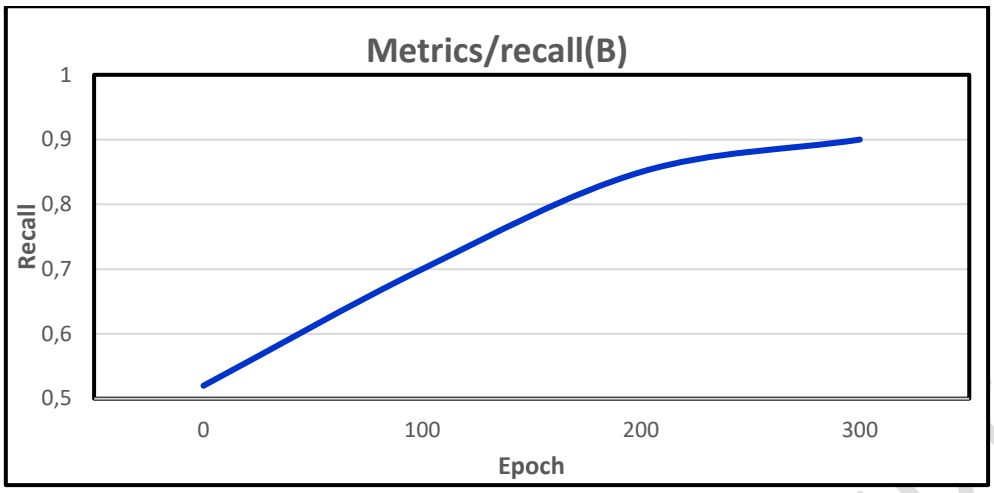
Table-2 Loss Value of the Proposed Model

Input Data	Box_Loss	Class_Loss	Distribution Focal_Loss
Training data	0.95	0.55	0.82
Validation data	0.98	0.20	0.80

In table-2 train/box loss indicates a steady decline during training, levelling off near 0.95 toward the final epochs. This reflects progressive improvement in the model's bounding-box regression capability. Likewise, the val/box loss shows a downward pattern, falling to roughly 0.85. Although the validation fluctuates more, this is typical and suggests the model maintains strong generalization without clear overfitting. Overall, both box-related losses demonstrate stable learning and enhanced localization performance. Similarly, the classification losses for both training and validation follow a consistent improvement trend throughout the 300-epoch process. The training classification loss reduces to about 0.55, indicating the model's increasing accuracy in class prediction on the training data. Meanwhile, the validation classification loss (val/cls loss) settles around 0.75, confirming reliable generalization and alignment between training and validation behavior. The Distribution Focal Loss (DFL) also confirm effective learning dynamics. The train/dfl loss gradually declines, reaching approximately 0.84, showing that the model becomes more precise in predicting object boundaries. The validation DFL loss starts near 1.0, with natural fluctuations, but continuously drops to around 0.83. The close proximity of training and validation DFL values further indicates that the model maintains generalization quality and avoids overfitting, supporting improved spatial prediction accuracy.



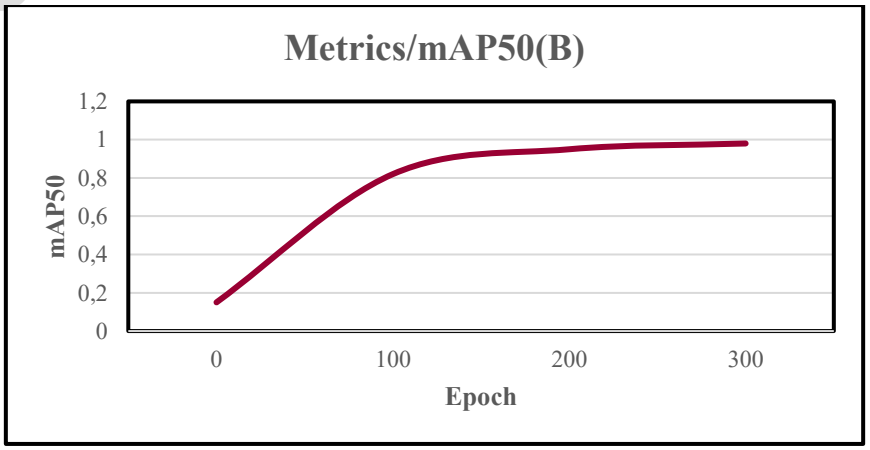
(a)



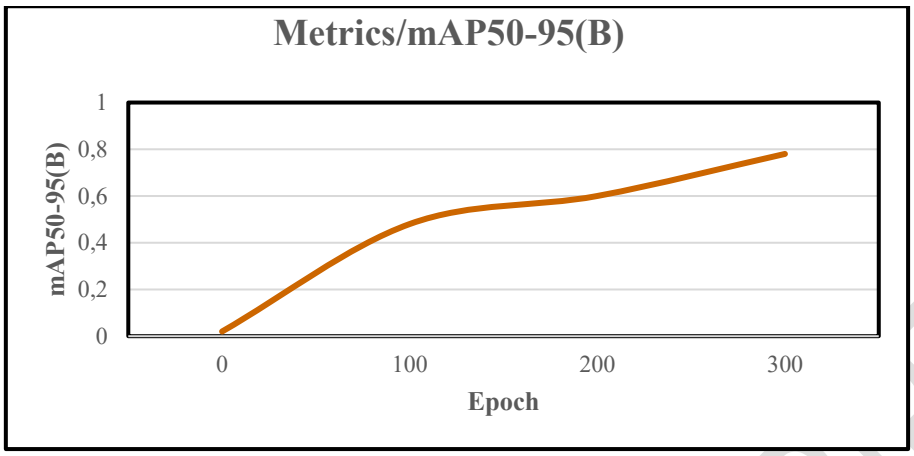
(b)

Figure-7 (A) Precision and **(B)** Recall Score

Figures 7 (a) and (b) illustrate precision and recall performance metrics over 300 training epochs. Figure 7(a) shows the precision metric (B), which measures the model's ability to predict positive detections correctly. The precision is initially low, at around 1.15, but after epoch 100, it increases suddenly and consolidates above 0.85, eventually attaining nearly 0.95 by the end of training. This indicates that the model becomes highly accurate at reducing false positives as training progresses. Figure 7(b) shows the recall metric (B), indicating the model's ability to detect all relevant instances. It starts at around 0.50 and steadily increases to about 0.90 by the end. This upward trend signifies that the model becomes more effective at detecting true objects in the dataset while minimizing false negatives over time. A well-balanced and dependable model relies on both precision and recall performance.



(a)



(b)

Figure-8 (a). mAP50**, (b)** mAP50-95 **(B)** Score

Figure 8 (A) illustrates the model's performance using the mean Average Precision (mAP) metric over 300 epochs. The Figure-(A) shows labeled metrics, including mAP50 (B), with the standard measure of detection quality, mAP, at an IoU threshold of approximately 0.50. Initially, it begins around 0.20 and quickly increases to around 0.95 by the end of training. This indicates that the model achieves high object localization and classification accuracy when evaluated at a lenient threshold. Figure 8 (B) shows the metrics of mAP50 (B), assessing performance across multiple IoU thresholds, which improves from 0.50 to 0.95. It starts at around 0.10, gradually rising and consolidating at 0.65 by epoch 300. This value reflects the model's true generalization performance, continuously increasing to assess the efficiency of its robust features for accurate object detection across varying levels of overlap.

(a)

(b)

Figure-9 (a) Input Image **(b)** predicted image

Figure-9 (A) and (B) depict the actual vs. predicted images of the proposed model, respectively. The results clearly demonstrate that the proposed model is more suitable for classifying hazardous and non-hazardous substances on the surface of floating water.

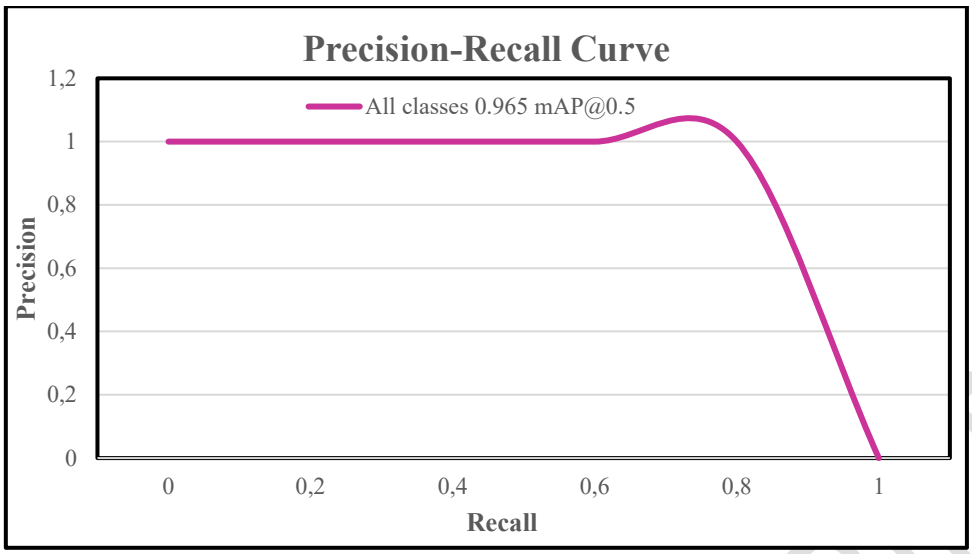


Figure-10 Precision-Recall Curve

The Precision-Recall (PR) curve in Figure 10 shows the model's effectiveness at detecting class 0. The curve remains close to the top-right corner, indicating that the model maintains high precision and recall across most thresholds. The model achieves a mean Average Precision at IoU 0.5 (mAP@0.5) of 0.965, demonstrating its accuracy in identifying the object class. A high score, nearing 1.0, indicates that the model strikes a balance between accurately identifying objects and minimizing errors, such as false positives or missed detections.

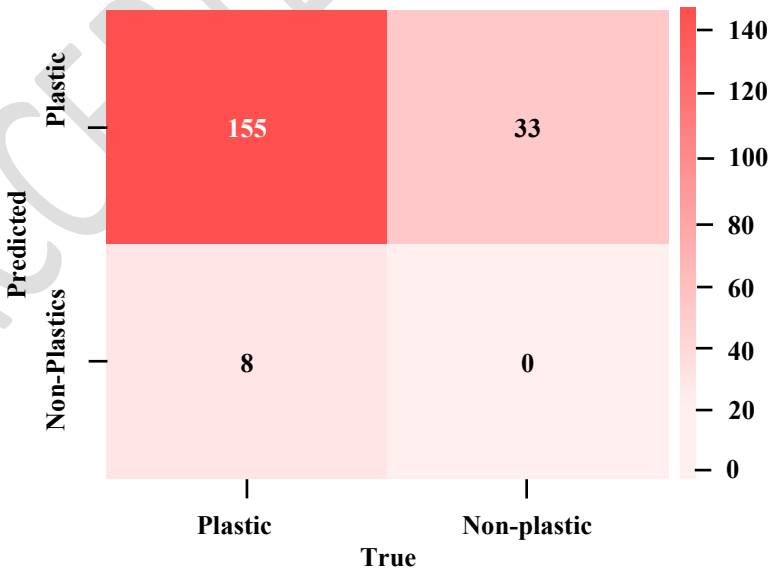


Figure 11: Confusion matrix

The model's binary classification is determined using a confusion matrix with classes 0 and background, as shown in Figure 11. This model at class 0 correctly predicted 155 instances; however, 33 background instances were misclassified, resulting in a high false-positive rate. Moreover, at class 0, 8 actual instances were misclassified as background. Given that 0 background instances were correctly identified, this indicates the model favors class 0 predictions and suggests the need to enhance balance by better training on the background class.

Table 3: Performance Comparison

Models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Inference Time (ms)**	**Alert delay (S)**
FRCNN [34]	91.6	91.2	89.9	211	NA
SSD [35]	86.9	86.9	84.0	59	NA
MobileNet [36]	79.2	79.2	78.7	35	NA
Proposed	93.8	93.8	92.5	48	2.1

The performance comparison results shown in Table-3 indicate that the proposed model outperforms the other models. In particular, the automatic alert generation module improves the model's efficiency and is more effective at detecting real-time pollutants in river water. Compared to other models, the proposed model is superior and faster, and it produces an alert with a 2.1-second delay. Additionally, the proposed model's detection accuracy is 93.8%, which is $\pm$2.4% to 18.4% higher than those of other models. Similarly, the precision score of the proposed model is 93.8%, and it is $\pm$2.85% to 18.4%, and the recall values is 92.51%, and it is $\pm$2.89% to 17.5% higher than those of the other models. These results demonstrate that the proposed model is well-suited for supporting environmental sustainability in smart cities.

4.1 Limitations of the Study

In addition to confirming the results, the study also faces challenges, such as the model's reliance on visual data collected from CCTV cameras and drones, which reduces its effectiveness in poor visibility conditions, such as heavy rain, fog, or nighttime, when sources are not easily accessible. The provided dataset, which is custom-built and diverse, may still lack sufficient representation of all possible pollutant types and surrounding scenarios. Furthermore, the system mainly targets surface-level pollutant detection and does not consider underwater or chemical pollutants, limiting its comprehensiveness. Therefore, deploying it across different geographic areas may require careful calibration and scaling.

5. Conclusion

This paper demonstrates how CNNs can effectively and reliably detect floating objects in river images, such as trash, debris, and other water-related items. Convolutional Neural Networks learn patterns and details directly from images, enabling the system to distinguish between floating objects and the water surface, even under challenging conditions such as lighting changes, reflections, or water movement. The CNN process involves steps such as feature extraction, object region proposal, region classification, and bounding box refinement to ensure accurate detection with minimal human intervention. This method benefits real-time environmental monitoring, waste detection in water, and tracking water quality. It supports pollution control, conservation, and smart water management initiatives. The model trained on the ReWater dataset showed a strong ability to identify plastic waste in rivers, achieving a high mean Average Precision (mAP) of 0.965 at an IoU of 0.5, demonstrating its effectiveness. The confusion matrix showed 155 correct identifications for class 0, with 8 misses and 33 false detections, suggesting good overall performance. The precision and recall graphs consistently showed high accuracy levels. Meanwhile, loss graphs for box, class, and dfl decreased steadily across 300 training cycles, reflecting effective learning. These results confirm that both the dataset and model are robust for real-world environmental monitoring.

References

1. Denchak, M. (2018). Water pollution: Everything you need to know. *Nat. Resour. Def. Counc. NY*.
2. Cory Ochs, Kaitlyn Garrison, Priyam Saxena, Kristen Romme, Atanu Sarkar, (2024), "Contamination of aquatic ecosystems by persistent organic pollutants (POPs) originating from landfills in Canada and the United States: A rapid scoping review", Science of The Total Environment, Vol. 924, No.171490,https://doi.org/10.1016/j.scitotenv.2024.171490.
3. Dwivedi, A. K. (2017). Researches in water pollution: A review. *International Research Journal of Natural and Applied Sciences*, *4*(1), 118-142.
4. Maharjan, N., Miyazaki, H., Pati, B. M., Dailey, M. N., Shrestha, S., & Nakamura, T. (2022). Detection of river plastic using UAV sensor data and deep learning. *Remote Sensing*, *14*(13), 3049. https://doi.org/10.3390/rs14133049
5. Siegfried, M., Koelmans, A. A., Besseling, E., & Kroeze, C. (2017). Export of microplastics from land to sea. A modelling approach. *Water research*, *127*, 249-257. https://doi.org/10.1016/j.watres.2017.10.011
6. Khan, A. S., Anavkar, A., Ali, A., Patel, N., & Alim, H. (2021). A review on current status of riverine pollution in India. *Biosciences Biotechnology Research Asia*, *18*(1), 9-22. http://dx.doi.org/10.13005/bbra/2893
7. Das, A. (2025). Applying the water quality indices, geographical information system, and advanced decision-making techniques to assess the suitability of surface water for drinking purposes in the Brahmani River Basin (BRB), Odisha. Environmental Science and Pollution Research, 1-36. https://doi.org/10.1007/s11356-025-36329-z
8. Das, A. (2024). Evaluation of surface water quality in Brahmani River Basin, Odisha (India), for drinking purposes using GIS-based WQIs, multivariate statistical techniques, and semi-variogram models. Innovative Infrastructure Solutions, 9(12), 484. https://doi.org/10.1007/s41062-024-01780-3

9. Gani, A., Hussain, A., & Pathak, S. (2025). Government Strategies to Economically Support the Cleaning of Water Bodies in India. *Wastewater to Resource Recovery: Applying the Circular Economy Toward Sustainable Development*, 575-596. https://doi.org/10.1002/9781394274314.ch25
10. Chatrabhuj, Meshram, K., Mishra, U., & Omar, P. J. (2024). Integration of remote sensing data and GIS technologies in the river management system. Discover Geoscience, 2(1), 67. https://doi.org/10.1007/s44288-024-00080-8
11. Ravish, P. Spatiotemporal Assessment of Water Quality in the Yamuna River and Its Tributaries in Haryana. http://dx.doi.org/10.12944/CWE.20.3.18
12. Fulazzaky, M. A. (2010). Water-quality evaluation system to assess the status and suitability of Citarum River water for various uses. *Environmental Monitoring and Assessment*, *168*, 669-684. https://doi.org/10.1007/s10661-009-1142-z
13. Patil, P. N., Sawant, D. V., & Deshmukh, R. N. (2012). Physico-chemical parameters for testing of water–A review. *International journal of environmental sciences*, *3*(3), 1194-1207.
14. Ogidi, O. I., & Akpan, U. M. (2022). Aquatic biodiversity loss: impacts of pollution and anthropogenic activities and strategies for conservation. In *Biodiversity in Africa: potentials, threats and conservation* (pp. 421-448). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-3326-4_16
15. Sonone, S. S., Jadhav, S., Sankhla, M. S., & Kumar, R. (2020). Water contamination by heavy metals and their toxic effect on aquaculture and human health through the food Chain. *Lett. Appl. NanoBioScience*, *10*(2), 2148-2166. https://doi.org/10.33263/LIANBS102.21482166
16. Demirbas, A. (2011). Waste management, waste resource facilities, and waste conversion processes. *Energy Conversion and Management*, *52*(2), 1280-1287. https://doi.org/10.1016/j.enconman.2010.09.025

17. Mostaghimi, K., & Behnamian, J. (2023). Waste minimization towards waste management and cleaner production strategies: a literature review. *Environment, Development and Sustainability*, *25*(11), 12119-12166. https://doi.org/10.1007/s10668-022-02599-7

18. Pihlajarinne, T. (2021). Repairing and re-using from an exclusive rights perspective: towards sustainable lifespan as part of a new normal?. In *Intellectual Property and Sustainable Markets* (pp. 81-100). Edward Elgar Publishing. https://doi.org/10.4337/9781789901351.00010

19. Parkinson, H. J., & Thompson, G. (2003). Analysis and taxonomy of remanufacturing industry practice. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, *217*(3), 243-256. https://doi.org/10.1243/095440803322328890

20. Khalid, I., Ullah, S., Umar, I. S., & Nurdiyanto, H. (2022). The problem of solid waste: origins, composition, disposal, recycling, and reusing. *International Journal of Advanced Science and Computer Applications*, *1*(1), 27-40. https://doi.org/10.47679/ijasca.v1i1.6

21. Chaaban, M. A. (2001). Hazardous waste source reduction in materials and processing technologies. *Journal of Materials Processing Technology*, *119*(1-3), 336-343. https://doi.org/10.1016/S0924-0136(01)00920-7

22. McShane, J., Meehan, K., Furey, E., & McAfee, M. (2021, December). Classifying plastic waste on river surfaces using CNNs and TensorFlow. In 2021, IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0475-0481). IEEE. https://doi.org/10.1109/UEMCON53757.2021.9666556

23. Sio, G. A., Guantero, D., & Villaverde, J. (2022, October). Plastic waste detection on rivers using the YOLOv5 algorithm. In *2022, the 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICCCNT54827.2022.9984439

24. van Lieshout, C., van Oeveren, K., van Emmerik, T., & Postma, E. (2020). Automated river plastic monitoring using deep learning and cameras. *Earth and space science*, *7*(8), e2019EA000960. https://doi.org/10.1029/2019EA000960

25. Faisal, M., Chaudhury, S., Sankaran, K. S., Raghavendra, S., Chitra, R. J., Eswaran, M., & Boddu, R. (2022). Faster R-CNN Algorithm for Detection of Plastic Garbage in the Ocean: A Case for Turtle Preservation. *Mathematical Problems in Engineering*, *2022*(1), 3639222. https://doi.org/10.1155/2022/3639222
26. Armitage, S., Awty-Carroll, K., Clewley, D., & Martinez-Vicente, V. (2022). Detection and classification of floating plastic litter using a vessel-mounted video camera and deep learning. Remote Sensing, 14(14), 3425. https://doi.org/10.3390/rs14143425
27. Yang, J., Li, Z., Gu, Z., & Li, W. (2024). Research on a floating object classification algorithm based on a convolutional neural network. Scientific Reports, 14(1), 32086. https://doi.org/10.1038/s41598-024-83543
28. Chellaiah, C., Anbalagan, S., Swaminathan, D., Chowdhury, S., Kadhila, T., Shopati, A. K., ... & Amesho, K. T. (2024). Integrating deep learning techniques for effective river water quality monitoring and management. *Journal of Environmental Management*, *370*, 122477. https://doi.org/10.1016/j.jenvman.2024.122477
29. Thavasimuthu, R., Vidhya, P. M., Sridhar, S., & Sherubha, P. (2024). SegNet-VOLO model for classifying microplastic contaminants in water bodies. *Polymers for Advanced Technologies*, *35*(7), e6497. https://doi.org/10.1002/pat.6497
30. Arepalli, P. G., & Naik, K. J. (2024). An IoT-based smart water quality assessment framework for aqua-ponds management using Dilated Spatial-temporal Convolution Neural Network (DSTCNN). *Aquacultural Engineering*, *104*, 102373. https://doi.org/10.1016/j.aquaeng.2023.102373
31. BENDIB, M. D. (2024). *Smart Waste Underwater Segmentation: Deep Learning Based Approach* (Doctoral dissertation, Echahid Chikh Larbi Tébessi University-Tébessa).
32. Hou, Q., Liu, Y., Sun, Z., Li, X., & Wei, J. Deep Learning-Based Autonomous Monitoring and Cleanup of Plastic Debris in Inland Waters. *Available at SSRN 5005153*. https://dx.doi.org/10.2139/ssrn.5005153

33. https://www.kaggle.com/datasets/abhranta/narmada-river-plastic-pollution-detection

34. Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031.

35. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single-shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2

36. Wang, Y., Li, S., Lin, Y., & Wang, M. (2021). Lightweight deep neural network method for water body extraction from high-resolution remote sensing images with multisensors. Sensors, 21(21), 7397. https://doi.org/10.3390/s21217397.

37. Venkatesan, A., & Krishnan, T. (2025). A hybrid cnn-lstm predictive model deployed federated learning model for advanced flood prediction systems to forecast coastal region of smart cities. *Global NEST Journal*, *27*(6).