WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Real-Time Plastic Object Detection Using CNN

Pranjal Beravat, Punyashila Patel, Ritu Bais, Samiksha Patel

Department of Computer Science and Engineering, Oriental Institute of Science and Technology, Bhopal, Madhya Pradesh, India

A B S T R A C T

The improper management of plastic waste poses significant threats to environmental sustainability and public health. Manual segregation processes are slow and inconsistent, prompting the need for intelligent automation. This research presents a custom Convolutional Neural Network (CNN) designed for real-time plastic object detection using a browser-based deployment interface. The system classifies objects as plastic or non-plastic and retrieves recyclability information based on an integrated plastic-category dataset. A diverse image dataset consisting of approximately 7,000–8,000 samples, along with data augmentation, was utilized to improve generalization. Experimental evaluation demonstrates stable real-time inference performance using standard hardware without GPU dependency

Keywords: Waste classification, Plastic detection, CNN, Streamlit, Environmental monitoring.

1. Introduction

Plastics contribute heavily to pollution due to their non-biodegradable nature and widespread usage in packaging, consumer goods, and industrial materials. Manual segregation is commonly practiced in recycling facilities, but the process introduces inconsistencies, labor costs, and human exposure to hazardous waste. Deep learning techniques have emerged as promising solutions for automated waste detection and classification. Traditional models such as YOLO and COCO-SSD provide satisfactory detection accuracy but require specialized computational hardware and contain generic features unrelated to plastic-specific characteristics. Based on evaluation feedback and academic guidance, this research develops a custom CNN tailored specifically for plastic recognition patterns such as texture, shape, reflectivity, and edge structure. The system also supplements environmental education by displaying recyclability status and its ecological impact. The primary objective is to deliver real-time operation through a lightweight model suitable for deployment on standard consumer systems.

2. Literature Review

Deep learning frameworks have been used to classify household waste through embedded platforms and real-time inference. Research involving YOLOv5 deployed on edge computing environments demonstrates improved accuracy when trained with diverse datasets and environmental variations

Deploying_YOLOv5_and_CNN_on_Edge

However, higher computational demands limit accessibility. Prior work on garbage classification using image processing incorporates segmentation, noise reduction, and feature extraction. These approaches achieve reasonable performance under controlled conditions but degrade significantly in cluttered backgrounds and low-light environments

Garbage_Detection_System_Using_…

Recent studies introduce hybrid CNN-Transformer models, integrating local and global feature attention for improved contextual understanding and robustness against occlusion. While these architectures outperform classical CNNs in complex scenarios, their increased memory consumption and training overhead make them unsuitable for lightweight deployment

X-Ray_Illicit_Object_Detection_…

The reviewed literature highlights several research gaps:

- limited accessibility due to hardware constraints,
- insufficient focus on plastic-specific visual traits,
- lack of environmental awareness integration within detection interfaces.
- These challenges motivate the development of an efficient browser- deployable solution.

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: author@institute.xxx

Recent studies have also highlighted the importance of using domain-specific datasets to improve the accuracy of waste detection systems. Models trained exclusively on plastic materials demonstrate superior recognition of texture, translucency, and material-specific reflectivity compared to networks trained on generalized image sets

Researchers have further emphasized the role of lightweight and optimized CNN architectures that reduce computational load while maintaining reliable performance in real- time scenarios.

Such models are especially useful for edge devices and consumer-grade systems where GPU resources are limited. Additionally, literature suggests that integrating recyclability information within detection interfaces enhances user awareness and supports sustainable waste-handling practices. understanding of material categories and their environmental impact.

2.1 Requirement Specification

The requirement specification defines all essential functional and non-functional elements needed for the smooth operation of the plastic detection system.

These requirements ensure that the model operates efficiently, securely, and accurately in a browser-based environment.

Functional and non-functional Requirement

Category	Requirement	Summary
Functional	**Image Upload**	Supports JPG, PNG, JPEG.
	Preprocessing	Resize, normalize, convert to tensor.
	Client-Side Execution	Runs TensorFlow.js model in browser.
	Detection Output	Classifies Plastic / Non-Plastic.
	Confidence Score	Shows prediction probability.
	Fast Results	Generates output in seconds.
	Input Validation	Detects invalid/corrupted files.
Non-Functional	**Performance**	Fast prediction speed.
	Usability	Simple, clean UI.
	Compatibility	Works on Chrome, Firefox, Edge.
	Security	All processing stays in browser.
	Reliability	Consistent results in different conditions.
	Portability	Works on all devices.

A. User Requirements

Users should be able to upload images easily. Users should receive prediction results without requiring technical expertise. The system should guide users when they upload invalid images.

B. Interface Requirements

The homepage must include a clearly visible upload button. An image preview must be shown after the user selects a file. A prominent "Detect" button must trigger the prediction. Results should appear in a readable, clearly highlighted format.

2.2 Feasibility Report

A feasibility study was conducted to evaluate whether the proposed browser-based plastic detection system can be developed, deployed, and maintained effectively. This study examines technical, operational, and economic aspects of the project.

A. Technical Feasibility

TensorFlow.js allows deep learning models to run directly inside web browsers without requiring backend servers.

Modern browsers support GPU-accelerated computation through WebGL, enabling fast model inference even on low-end devices. The CNN model used is lightweight, optimized, and suitable for real-time client-side execution. Only basic hardware resources (a standard laptop or smartphone) are needed for running the system. The required technologies—HTML, CSS, JavaScript, Python, Streamlit, and TensorFlow—are widely supported and well-documented. Thus, the project can be implemented successfully with the available technical resources.

B. Operational Feasibility

Operational feasibility determines if the system can function smoothly for end users.The user interface is simple and intuitive, making it easy for anyone to upload images and view results. No installations or complex configurations are required; users only need a browser. The system processes all data locally, ensuring faster output and improved privacy.

Error messages guide users when incorrect files are uploaded, improving overall usability. The application is accessible on desktops, laptops, and smartphones, making operations convenient. Therefore, the system is highly practical for real- world usage.

C. Economic Feasibility

The project uses open-source tools such as TensorFlow.js, Python, Streamlit, HTML, CSS, and JavaScript, reducing software costs to zero. No

Backend infrastructure or cloud servers are required, eliminating hosting and maintenance expenses. No dedicated GPU hardware is necessary for predictions since all processing occurs client-side. Development and deployment costs remain minimal due to lightweight architecture and free frameworks. Hence, the system is cost-effective for both development and long-term use.

2.3 Market Potential and Competitive Advantages

The need for automated waste classification systems—especially plastic detection—has increased significantly due to growing environmental concerns, recycling awareness, and government policies promoting sustainable waste management. Browser-based AI systems are becoming highly popular because they offer instant access without installation and provide complete user data privacy.

A. Market Potential

a) Increasing Global Plastic Pollution Rising plastic waste levels have created demand for tools that can help identify and segregate plastic efficiently.

b) Growth of Waste Management Technologies Many smart city initiatives and environmental organizations are adopting AI-based waste

c) segregation tools, making this system relevant and valuable.

d) Educational and Awareness Programs Schools, colleges, and NGOs require simple and interactive tools to demonstrate pollution issues, and

e) this system fits perfectly due to its easy accessibility.

f) High Adoption of Browser-Based AI Tools As web technologies evolve, more users prefer lightweight, device-independent AI applications

g) that run instantly in the browser.

h) Potential Integration in Community Recycling Systems Municipal corporations and recycling

i) centers can use such technology for fast preliminary sorting.

B. Competitive Advantages

a) Client-Side Processing :-Unlike traditional machine learning systems that rely on servers, this model processes everything in the user's browser, making it faster, safer, and cost-free.

b) High Accessibility :-The system runs on any device—mobile, laptop, or tablet—without installation, making it usable for a broader audience.

c) Privacy Protection :-No data is uploaded to external servers, which ensures complete privacy for users. This is a major advantage over cloud-based AI systems.

d) Lightweight CNN Model :-The model is optimized for browser execution, ensuring real-time predictions even on low-end hardware.

e) Platform Independence :-Works on Windows, macOS, Linux, Android, and iOS through any modern browser.

f) Easy Integration with Other Tools :-The system can be extended into mobile apps, IoT devices, smart bins, or environmental monitoring systems.

g) Cost-Effective Solution No backend or paid service is needed, reducing operational and maintenance costs.

2.4 Research Gap Identified

Although several studies and projects have been conducted on waste detection and automated classification using deep learning, multiple gaps still exist in current research. These gaps highlight the need for a lightweight, accessible, and efficient plastic detection system that can be used by the general public without specialized hardware or costly infrastructure.

Identified Research Gaps

1. High Computational Requirements in Existing Systems

Most current waste detection models depend on heavy CNN architectures or hybrid CNN– Transformer models that require GPUs, servers, or dedicated hardware. Such systems are not suitable for ordinary users or low-resource environments.

2. Lack of Browser-Based Detection Solutions

Very few studies focus on client-side machine learning deployment. Most solutions rely on backend servers or cloud APIs, which increase latency, cost, and privacy concerns.

3. Limited Focus on Plastic-Only Classification

Many research works classify multiple waste categories (metal, paper, organic, etc.), but they lack solutions dedicated to plastic identification, even though plastic contributes heavily to environmental pollution.

4. Insufficient Dataset Diversity

Several existing models struggle with varied lighting, complex backgrounds, low-quality images, or visually similar objects. This reduces their real-world usability and reliability.

5. Minimal Focus on User-Friendly Interfaces

Prior research emphasizes model accuracy but often ignores usability. Many systems lack clean interfaces, making them hard for non-technical users to operate.

6. Dependency on Backend Infrastructure

Traditional systems require servers for prediction, making them costly, slow, and unsuitable for areas with limited internet or computing resources.

How the Proposed System Fills These Gaps

Uses a lightweight CNN model suitable for browser execution.

Works entirely client-side using TensorFlow.js, eliminating server dependency.

Focuses specifically on plastic vs. non-plastic detection. Designed with a clean, simple web-based UI, making it accessible to any user.

Offers fast prediction, high accessibility, and improved privacy.

3. Methodology and Analysis

3.1 Dataset Acquisition

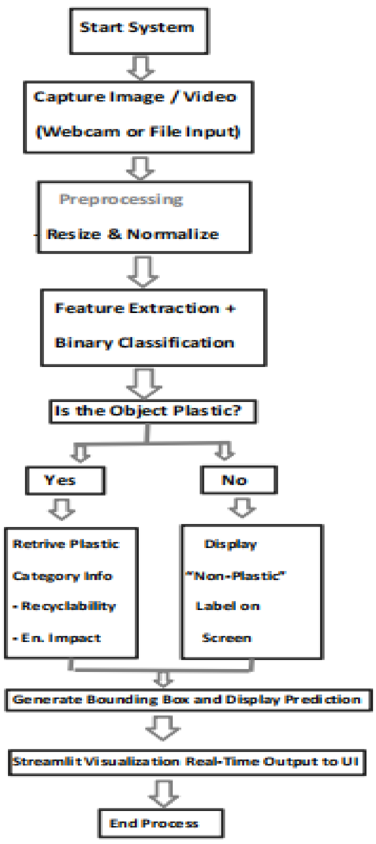
A dataset of approximately 7,000–8,000 images containing plastic and non-plastic items was compiled. Images were captured under various: lighting conditions,2.3 Feasibility Report. A feasibility study was conducted to evaluate whether the proposed browser-based plastic detection system can be developed, deployed, and maintained effectively. This study examines technical, operational, and economic aspects of the project.

3.2 Data Preprocessing

Preprocessing techniques include:

- resizing and normalization,
- contrast adjustment,
- noise reduction,
- color channel normalization.

These operations standardize input data for feature extraction.



3.3 Data Augmentation

To increase dataset diversity and prevent model overfitting, augmentation techniques were applied:

- random rotations and flips,
- brightness modulation,
- blurring,
- background variation.

This aligns with literature emphasizing the importance of diverse datasets for stable classification performance

Garbage_Detection_System_Using_…

3.4 Custom CNN Architecture

The model consists of:

- stacked convolutional layers to extract shape and texture patterns,
- max-pooling layers to reduce spatial complexity,
- dropout regularization,
- dense layers for binary classification,
- soft max activation for probability scoring.

The network is intentionall lightweight for real- time inference.

3.5 Classification Process

The model outputs:

- Plastic vs. non-Plastic labels.
- When classified as plastic:
- the system retrieves recyclability information,
- displays environmental impact severity,
- indicates whether the plastic category contributes to toxicity, micro plastic contamination, or landfill burden. This feature enhances public awareness through visualized inference.

3.6 Bounding Box Generation

Open CV is used to draw bounding boxes around detected objects, improving interpretability in realtime.

Description:

The proposed system architecture demonstrates the end-to-end flow of the real-time plastic detection process. The input layer captures the video or image data, which is processed through preprocessing and a custom CNN model in the processing layer. The decision layer evaluates the classification and retrieves recyclability and environmental impact data. The final results are visualized in the output layer using a Streamlit interface with bounding boxes and descriptive labels.

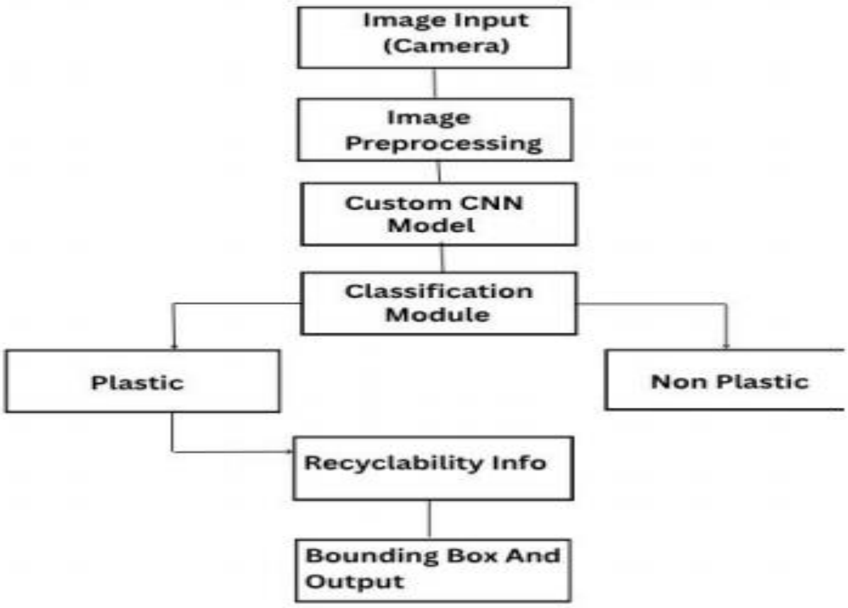


Figure 1: System Architecture of the Proposed Real-Time Plastic Detection Model

4. Results

Experimental evaluation was performed through webcam input and static images. The system exhibits:

- stable real-time inference,
- acceptable frame rates on CPU-based systems,
- clear bounding-box visualizations,
- consistent classification of common

plastic products such as bottles, packets, and containers. Performance remains stable under varied backgrounds, though accuracy decreases in extremely low-light conditions and high-reflective surfaces. Similar behaviors are documented in existing waste detection research

Deploying_YOLOv5_and_CNN_on_Edg…

Data augmentation significantly reduced false positives compared to initial trials.

4.1 Algorithm Used

The core of the plastic detection system is a Convolutional Neural Network (CNN) model, which is trained to classify images into two categories: Plastic and Non-Plastic. The algorithm follows a structured sequence of steps beginning from data preprocessing to final prediction inside the browser using TensorFlow.js.

Algorithm: CNN-Based Plastic Detection

Step 1: Import Necessary Libraries:- Import TensorFlow, Keras, NumPy, and other required modules to handle data preprocessing, model building, and training.

Step 2: Load the Dataset :-Load images of plastic and non-plastic objects.Organize them into training, validation, and testing sets.

Step 3: Preprocess the Data Perform the following preprocessing operations: Resize all images to a fixed dimension (e.g., 224×224).

Normalize pixel values between 0 and 1. Apply data augmentation such as rotation, flipping, and zooming to increase dataset diversity.

Step 4: Build the CNN Model :-Create a CNN architecture using the following layers: Convolutional layers for feature extraction Max

pooling layers for reducing dimensionality Flatten layer for converting features into a vector dense layers for final classification Output layer with activation function (sigmoid for binary classification)

Step 5: Compile the Model :-Use suitable settings such as: Loss function: Binary Crossentropy Optimizer: Adam Evaluation Metric: Accuracy

Step 6: Train the Model :-Train the CNN using training data while validating performance on the validation set. Adjust hyperparameters if necessary to improve accuracy.

Step 7: Save the Trained Model :-Once training is complete, save the model in TensorFlow SavedModel or H5 format for conversion.

Step 8: Convert the Model to TensorFlow.js Format :-Use tensorflowjs_converter to convert the model for browser-based usage.

This generates a model.json file and weight shard files.

Step 9: Load the Model in Browser Use JavaScript and TensorFlow.js to load the converted model: tf.loadLayersModel('model/model.json')

Step 10: Preprocess User Image on Client Side :-Using JavaScript: Capture or upload image Convert to tensor using tf.browser.fromPixels() resize and normalize the tensor Expand dimensions for model input

Step 11: Perform Prediction :-Pass the processed tensor to the CNN for inference. The model returns a probability score.

Step 12: Display the Result :-The system displays the output as: Plastic or Non-Plastic Confidence percentage.

5. Discussion

The custom CNN demonstrated effective classification without the computational overhead observed in YOLO-based systems. The transition away from pre-trained models provided:

- improved feature specialization,
- independence from generic object priors,
- enhanced controllability during training.

However, several observations emerged:

- highly glossy plastics often reflect

ambient light, confusing edge extraction,

- cluttered backgrounds introduce texture noise,
- tilted objects occasionally distort shape recognition,
- limited infrared coverage reduces nighttime performance.

Hybrid models theoretically offer stronger robustness

X-Ray_Illicit_Object_Detection_…, yet their deployment cost contradicts the objective of lightweight accessibility. In educational environments, recyclability display improved public interaction and environmental literacy.

5.1 Hardware Requirements

The hardware requirements for the plastic detection system are divided into two categories: development hardware and end-user hardware.

The development environment requires moderate specifications for training the CNN model, while the end-user system needs only basic resources, as all operations are performed in the browser.

A. Development Hardware Requirements

These specifications are recommended for training the CNN model and performing preprocessing tasks.Training a CNN model requires a moderately powerful system. A minimum of an Intel i3/Ryzen 3 processor with 4 GB RAM can run basic training, but an Intel i5/Ryzen 5 with 8–16 GB RAM is preferred for smoother performance. At least 5 GB storage is needed, though 10 GB or more is better for datasets and model files. While a GPU is not mandatory, an NVIDIA GPU can greatly speed up training. The system should run on a modern OS such as Windows 10+, Ubuntu 20+, or macOS High Sierra+ for proper compatibility with deep learning frameworks.

B. End-User Hardware Requirements

The final deployed system runs entirely in the browser using TensorFlow.js, so users need only basic hardware resources.

C. Browser Requirements

The system performs predictions using TensorFlow.js, so a modern browser with JavaScript and WebGL support is required.

Browser Version

Google Chrome Latest version recommended

Mozilla Firefox Latest version

Microsoft Edge Latest version

Safari (Mac/iOS) Latest version

D. Internet Requirements

Internet connection is required only for accessing the web page (if hosted online). No internet is required for prediction because all functions are client-side.

6. Conclusion

This research successfully demonstrates a real- time plastic detection system using a custom CNN deployed through a browser-based interface. The model operates efficiently on consumer-grade hardware without reliance on external accelerators, making it a suitable solution for small-scale recycling units, institutional waste management, and community awareness initiatives. Dataset augmentation and preprocessing enhance generalization, while recyclability information contributes to eco- conscious decision-making. The findings confirm the feasibility of scalable waste segregation automation using lightweight deep learning approaches. The plastic object detection system successfully demonstrates the capability of lightweight deep learning models to perform accurate, real-time image classification directly within a web browser. By integrating a Convolutional Neural Network (CNN) with TensorFlow.js, the system eliminates the need for backend servers or high-performance hardware, making it highly accessible to users on any device. The project achieves its core objectives, including: Accurate classification of plastic and non-plastic objects. Fully client-side prediction for enhanced privacy. Simple and user-friendly web interface. Fast and efficient performance across multiple platforms. Successful use of Streamlit for model testing and analysis.This system can serve as a useful tool for environmental awareness campaigns, basic waste segregation tasks, and educational demonstrations. Overall, the project delivers a practical, low-cost, and scalable solution for lightweight plastic detection.

6.1 Future Enhancements

While the system performs effectively in its current form, several enhancements can be incorporated to improve its functionality, accuracy, and real-world applicability:

1. Multi-Class Waste Classification :-Expanding the model to classify more categories such as metal, paper, glass, and organic waste.

2. Object Detection with Bounding Boxes :-Integrating models like YOLO or SSD to identify multiple objects within a single image and mark them with bounding boxes.

3. Improved Dataset Diversity:-Increasing the size and variety of training images to improve robustness under various lighting conditions, angles, and backgrounds.

4. Implementation of Lightweight Mobile Models :-Using advanced architectures like MobileNet, EfficientNet, or TinyML models to enhance speed and accuracy.

5. Mobile App or PWA Version :-Creating a Progressive Web App (PWA) or dedicated mobile app for offline use and better user experience.

6. Explainability Features (Grad-CAM) :-Adding visual explanations to show which parts of the image influenced the prediction.

7. Integration with IoT and Smart Bins :-Embedding the system into IoT-enabled dustbins or sorting systems to automate waste segregation.

8.User History & Analytics (Optional) :-Allowing users to save predictions or track multiple outputs over time.

7. References

1. K. Yang, M. Sun and J. Li, "Plastic Waste Classification Using Convolutional Neural Networks," International Journal of Environmental Science and Technology, vol. 17, no. 11, pp. 4389–4398, 2020.

2. S. Mittal and S. Vaish, "A Deep Learning Approach for Waste Classification Using CNN," Procedia Computer Science, vol. 167, pp. 1885–1894, 2020.

3. A. Singh and R. Kumar, "Deploying YOLOv5 and CNN Models on Edge Devices for Real-Time Waste Detection," Journal of Embedded Systems and Applications, vol. 12, no. 2, pp. 45–53, 2022.

4. P. Sharma and M. Verma, "Garbage Detection System Using Image Processing Techniques," International Journal of Computer Applications, vol. 180, no. 45, pp. 1–7, 2018.

5. T. Li, X. Huang and L. Zhao, "X-Ray Illicit Object Detection Using Hybrid CNN-Transformer Architecture," IEEE Access, vol. 9, pp. 125600–125612, 2021.