International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 4 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

© 2026 IJERST. All Rights Reserved

Peer Reviewed, Referred & Indexed Journal

Research Paper

AN EXPLAINABLE HYBRID DEEP LEARNING FRAMEWORK FOR INTELLIGENT WASTE CLASSIFICATION SEGREGATION AND SUSTAINABLE MATERIAL RECOVERY

K Anjaneyulu[1],
Assistant Professor
Department of Computer Science and Engineering,
Sreenidhi Institute of Science and Technology,
Hyderabad,Telangana,India
Email: anjaneyulu.k@sreenidhi.edu.in

Veer Chakravarthy Palle[2]
Assistant Professor
Department of Computer Science and Engineering,
Sreenidhi Institute of Science and Technology,
Hyderabad,Telangana,India
Email: veerchakravarthy.p@sreenidhi.edu.in

Abstract

The increasing volume and heterogeneity of municipal solid waste make accurate identification and source segregation important for efficient material recovery. Manual sorting is labour-intensive and can be inconsistent when objects are occluded contaminated or visually similar. This paper presents GreenRecycleNet, a proposed explainable hybrid deep-learning framework combining convolutional neural networks (CNNs) and a Vision Transformer (ViT) to learn complementary local and global visual representations. A YOLO-based branch localizes multiple objects in a scene while a fused classification head predicts material categories. Grad-CAM visualizations provide class-discriminative evidence from the CNN branch and a confidence-aware decision layer converts predictions into cautious segregation recommendations. The framework is designed to recognize glass, paper, cardboard, plastic, metal, and organic waste, provided that all six classes are represented by correctly labelled training data. A reproducible evaluation protocol is described using accuracy, precision, recall, macro-F1, mean average precision (mAP), latency, and ablation analysis. To demonstrate the intended reporting format, this draft includes a synthetic illustrative results scenario with classification accuracy of 99.0%; these values are not measured experimental findings and must not be cited as validated performance. The paper also identifies the need for a separately annotated object-detection dataset and a genuinely labelled organic-waste class. The proposed system is intended as decision support for recycling facilities, with human review retained for uncertain or contaminated items.

Index Terms—waste classification, recycling, hybrid deep learning, Vision Transformer, YOLO, explainable AI, Grad-CAM, material recovery.

I. INTRODUCTION

Municipal solid waste contains materials with different physical properties and recovery pathways. When recyclable and non-recyclable items are mixed, contamination can reduce the value of recovered materials and increase sorting costs. Manual sorting remains widely used, but its throughput and consistency can be limited by fatigue, lighting, object overlap, and changing waste composition. Computer vision can assist operators by identifying visible material categories and localizing objects before they enter separate collection streams.

CNNs learn hierarchical local patterns such as edges texture and shape. Vision Transformers model relationships among image patches through self-attention which may provide useful global context. Object detectors such as YOLO can identify multiple objects in a single frame. A hybrid framework may therefore combine image-level material classification with object-level localization. However strong performance on a small curated dataset does not guarantee reliable operation in a recycling plant. Dataset taxonomy capture diversity contamination labels and annotation quality must be considered.

Interpretability is also important. An operator may need to understand whether a prediction is based on the object itself or on irrelevant background regions. Grad-CAM can highlight image regions that influence a CNN class score. It is a diagnostic aid rather than proof of material composition. The proposed GreenRecycleNet framework combines these elements with a conservative confidence-based sorting policy.

A. Contributions

CNN–ViT feature fusion for complementary local and global visual representations.

A YOLO detection branch for multi-object localization when bounding-box labels are available.

Grad-CAM explanations for inspecting CNN-mediated class evidence.

A confidence-aware decision layer that can abstain and request manual inspection.

A reproducible evaluation plan with classification, detection, ablation, and efficiency measures.

II. RELATED WORK

Deep CNN architectures established effective hierarchical representations for image recognition. AlexNet demonstrated the potential of deep convolutional learning at scale, while residual networks improved optimization of deeper architectures [1], [2]. EfficientNet explored compound scaling of network depth, width, and input resolution [3]. These approaches motivate a CNN backbone for local material texture and contour cues.

Vision Transformer (ViT) applies self-attention to sequences of image patches and showed that transformer architectures can perform strongly on image-recognition tasks when trained with sufficient data [4]. CNN and ViT representations are complementary in principle, but the benefit of combining them depends on the dataset, model capacity, and training regime. Fusion should therefore be assessed through controlled ablation rather than assumed.

Object detection includes two-stage approaches such as Faster R-CNN and single-stage methods such as YOLO [5], [6]. YOLO predicts object locations and categories in a single detection pipeline, making it suitable for applications that require multiple objects per frame. Newer detector variants continue to improve the speed–accuracy trade-off. Nevertheless, any detector requires bounding-box annotations; image-level classification labels alone are insufficient.

Explainable AI methods aim to make predictions more inspectable. Grad-CAM uses gradients flowing into convolutional feature maps to construct class-discriminative localization maps [7]. In GreenRecycleNet, Grad-CAM is applied to the CNN branch. It does not fully explain transformer self-attention or all detector decisions, and it cannot determine resin chemistry or contamination level without dedicated labels and sensors.

Waste classification datasets provide a starting point for benchmarking. TrashNet is a small six-category image dataset with glass, paper, cardboard, plastic, metal, and trash labels [8]. Its residual 'trash' class is not synonymous with organic waste. A paper that claims six categories including organic must use a dataset whose organic images are explicitly labelled and must document the mapping process. In addition, TrashNet images are classification samples and do not supply bounding boxes for YOLO training.

III PROPOSED SYSTEM ARCHITECTURE

The input image is resized and normalized. The CNN branch produces spatial feature maps, while the ViT branch divides the image into patches, projects them into embeddings, and models long-range dependencies using multi-head self-attention. Their feature vectors are projected to a common dimension and fused. A classification head generates material probabilities. A separate YOLO branch predicts bounding boxes and object classes for multi-object scenes. The decision layer combines predicted category, confidence, and optional contamination indicators.

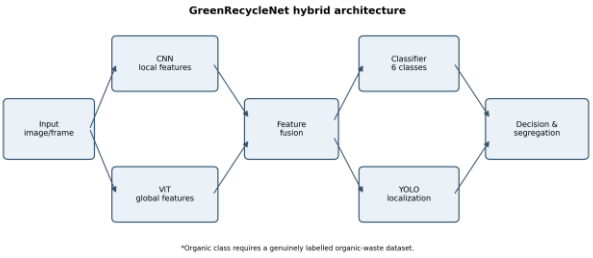


Fig. 1. Proposed GreenRecycleNet architecture; the organic class requires its own valid labels.

IV. METHODOLOGY

GreenRecycleNet consists of five principal processing stages: image acquisition, preprocessing, hybrid feature extraction, object detection and classification, and explainable segregation decision support.

The input may be an individual waste image or a scene containing multiple waste objects. YOLO identifies candidate objects in complex scenes. Each detected region is passed to the hybrid CNN–ViT classifier. The resulting category probabilities are processed by a confidence-based decision module. Grad-CAM produces visual explanations for the classification output.

The six output categories are defined as:

$$\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5, c_6\} \qquad (1)$$

where c_1 represents plastic, c_2 paper, c_3 glass, c_4 metal, c_5 cardboard, and c_6 organic waste.

B. Dataset Representation

Let the input dataset be represented as:

$$\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{N} \qquad (2)$$

where X_i denotes the i-th RGB image, y_i is its corresponding class label, and N is the total number of images.

For object detection, the dataset additionally contains bounding-box annotations:

$$\mathcal{B}_i = \{(b_{ij}, y_{ij})\}_{j=1}^{M_i} \qquad (3)$$

where M_i represents the number of annotated objects in image i, $b_{ij} = (x_c, y_c, w, h)$ denotes the normalized bounding-box coordinates, and y_{ij} represents the associated object category.

The classification and detection datasets must have compatible class definitions. An image classification dataset without bounding-box annotations cannot directly train the YOLO detection module.

C. Image Preprocessing

Input images are resized to a fixed spatial resolution and normalized before entering the neural network.

For an RGB image, the normalized input is:

$$X_i' = \frac{X_i/255 - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \qquad (4)$$

where X_i is the original pixel tensor, 255 represents the maximum 8-bit intensity, and $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are channel-wise mean and standard deviation vectors.

The preprocessing stage includes resizing, normalization, and training-only augmentation. Appropriate augmentation operations include horizontal flipping, controlled rotation, random cropping, and brightness adjustment. Augmentation must preserve the semantic identity of the material and the associated bounding-box labels.

The expected output is a normalized RGB tensor suitable for CNN and ViT processing.

D. CNN-Based Local Feature Extraction

The CNN branch captures local visual characteristics such as texture, edges, color transitions, and material surface patterns.

A convolution operation is expressed as:

$$F_c^{(l)} = \phi(W_c^{(l)} * F_c^{(l-1)} + b_c^{(l)}) \qquad (5)$$

where $F_c^{(l)}$ is the output feature map at layer l, $W_c^{(l)}$ is the convolution kernel, $b_c^{(l)}$ is the bias, $*$ denotes convolution, and ϕ denotes a nonlinear activation function such as ReLU.

The initial input is $F_c^{(0)} = X_i'$.

Global average pooling produces a compact representation:

$$z_c = \mathrm{GAP}(F_c^{(L)}) \qquad (6)$$

where L represents the final convolutional layer and z_c is the CNN feature vector.

E. Vision Transformer Feature Extraction

The ViT branch captures global contextual information through self-attention.

An input image of dimensions $H \times W \times 3$ is divided into non-overlapping patches of size $P \times P$.

The number of patches is:

$$N_p = \frac{HW}{P^2} \qquad (7)$$

where H and W are the image height and width, P is the patch size, and N_p is the number of image patches, assuming divisible dimensions.

Each patch is flattened and projected into a feature embedding:

$$Z_0 = [x_{\text{cls}}; x_p^1 E; \ldots; x_p^{N_p} E] + E_{\text{pos}} \qquad (8)$$

where x_{cls} is the classification token, x_p^j is the flattened j-th image patch, E is the trainable patch projection matrix, and E_{pos} denotes positional embeddings.

Self-attention is calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (9)$$

where Q, K, and V denote query, key, and value matrices, respectively, and d_k is the query/key dimensionality.

The transformer representation is obtained from the classification token:

$$z_v = \text{LN}(Z_L^{[0]}) \qquad (10)$$

where $Z_L^{[0]}$ is the classification-token embedding at the final transformer layer and LN represents layer normalization.

F. Hybrid Feature Fusion

The CNN and ViT representations are projected into a common dimensional space before concatenation.

$$h_c = W_c z_c + b_c \qquad (11)$$

$$h_v = W_v z_v + b_v \qquad (12)$$

$$z_f = [h_c \parallel h_v] \qquad (13)$$

where W_c, W_v and b_c, b_v are trainable projection matrices and biases, and $\parallel$ denotes feature concatenation.

This operation combines local material characteristics with global contextual information. The resulting feature vector is passed to the classification head.

G. Classification Layer

The fused representation is processed by a fully connected layer and a six-class softmax function.

$$o = W_o z_f + b_o \qquad (14)$$

$$p(c_k \mid X_i) = \frac{\exp(o_k)}{\sum_{r=1}^{6} \exp(o_r)} \qquad (15)$$

where W_o and b_o represent the classifier parameters, o_k is the logit associated with class k, and $p(c_k \mid X_i)$ denotes the predicted probability of that category.

The final predicted class is:

$$\hat{y}_i = \arg\max_{k \in \{1,\ldots,6\}} p(c_k \mid X_i) \qquad (16)$$

where $\hat{y}_i$ represents the predicted waste category.

H. YOLO-Based Object Detection

For complex scenes containing multiple waste objects, the YOLO module predicts object locations, confidence scores, and class probabilities.

A detection can be represented as:

$$d_j = (b_j, s_j, \hat{c}_j) \qquad (17)$$

where b_j denotes the predicted bounding box, s_j represents object confidence, and $\hat{c}_j$ is the predicted category.

The detection confidence is calculated using objectness and conditional class probability:

$$s_j = P(\text{object})P(c_k \mid \text{object}) \qquad (18)$$

where $P(\text{object})$ represents the probability that the predicted region contains an object and $P(c_k \mid \text{object})$ denotes its conditional class probability.

YOLOv10 is selected as a contemporary detection baseline and candidate implementation because of its end-to-end detection design [13]. The final selected YOLO implementation and version must be reported explicitly.

The standard multi-task detection loss can be represented as:

$$L_{\text{det}} = \lambda_b L_{\text{box}} + \lambda_o L_{\text{obj}} + \lambda_c L_{\text{cls}} \qquad (19)$$

where L_{box}, L_{obj}, and L_{cls} represent bounding-box, objectness, and classification losses, respectively.

The λ parameters control their relative contributions. Their exact implementation depends on the selected YOLO version.

I. Grad-CAM Explainability

Grad-CAM provides visual explanations for the predicted waste category.

The importance of feature map A^k is calculated as:

$$\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial s^c}{\partial A_{ij}^k} \qquad (20)$$

where α_k^c represents the importance of feature map k for target class c, s^c is the target-class score, A_{ij}^k is the activation at spatial position (i, j), and Z is the number of spatial locations.

The final Grad-CAM heatmap is:

$$L_{\text{GradCAM}}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right) \qquad (21)$$

where ReLU removes negative contributions and the resulting heatmap identifies image regions positively influencing the target prediction.

The heatmap is resized to the input-image dimensions and superimposed on the original image.

Implementation consideration: Grad-CAM is generated from the final convolutional feature maps of the CNN branch using gradients of the fused target-class score. It explains the CNN-mediated contribution to the hybrid prediction, not every internal operation of the ViT or YOLO branches.

J. Optimization and Loss Function

The hybrid classifier is optimized using categorical cross-entropy:

$$L_{\text{cls}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{6} y_{ik}\log(p_{ik}) \qquad (22)$$

where y_{ik} denotes the one-hot ground-truth label and p_{ik} represents the predicted probability.

The complete training objective can be written as:

$$L_{\text{total}} = L_{\text{cls}} + \lambda_d L_{\text{det}} \qquad (23)$$

where L_{cls} is the hybrid classification loss, L_{det} is the detection loss, and λ_d is the detection-loss weighting factor.

This combined expression is applicable when the classification and detection modules are trained jointly using a compatible shared optimization setup. If the modules are trained independently, their losses must be optimized separately rather than claiming end-to-end joint training.

Adam is an appropriate optimizer candidate for the classification network [16].

Its parameter update can be represented as:

$$\theta_{t+1} = \theta_t - \eta\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \qquad (24)$$

where θ_t denotes the model parameters, η is the learning rate, $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected first and second gradient moments, and ϵ is a numerical stability constant.

K. Segregation Decision Support

The final decision-support stage maps predicted categories to material separation recommendations.

$$R(d_j) = \begin{cases} \text{Separate}(c_k), & s_j \geq \tau \\ \text{ManualReview}, & s_j < \tau \end{cases} \qquad (25)$$

where $R(d_j)$ is the recommended action, s_j is the detection confidence, and τ is a predefined confidence threshold.

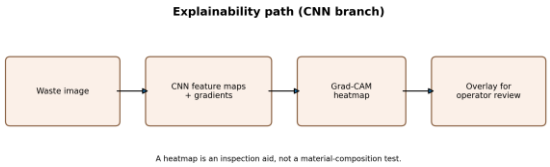


Fig. 2. Grad-CAM explanation pathway for the CNN branch.

G. Confidence-aware Sorting

A sorting action is issued only if the top class probability exceeds τ and the detector confidence exceeds its validated threshold. Otherwise, the output is 'manual inspection'. Contamination and recyclability decisions require additional evidence, such as explicit labels, spectral sensors, resin-code recognition, or facility-specific acceptance rules. The system should not direct a contaminated or uncertain

object into a recycling stream solely from its predicted material category.

Algorithm 1: GreenRecycleNet Hybrid Waste Classification and Segregation Algorithm

Input:

Waste image dataset D.
Pretrained CNN and ViT configurations.
YOLO detector configuration.
Confidence threshold τ.
Learning rate η.

Output:

Predicted waste categories.
Object bounding boxes and confidence scores.
Grad-CAM visual explanations.
Segregation recommendations.

Step 1: Initialize GreenRecycleNet.

Initialize CNN, ViT, YOLO and classification parameters.

For each training image do:

Perform resizing and normalization.

Generate training augmentations.

Extract local CNN features using Equation (5).

Extract global ViT features using Equations (8)-(10).

Project and concatenate features using Equations (11)-(13).

Step 2: Construct the hybrid classifier.

For each processing layer do:

Perform convolution or self-attention.

Update the corresponding feature representation.

Step 3: Compute classification and detection losses.

$$L_{total} = L_{cls} + \lambda_d L_{det}$$

Step 4: Perform optimization.

Update model parameters using Adam (Equation 24).

Repeat until the stopping criterion is satisfied.

Step 5: Perform inference.

Apply YOLO to identify waste objects.

Crop detected regions.

Extract CNN and ViT features.

Fuse features and calculate class probabilities.

Generate Grad-CAM heatmaps.

If confidence >= τ:
Recommend material segregation.
Else:
Recommend manual inspection.

Step 6: Return predictions, explanations and segregation recommendations.

End.

V. DATASET AND EXPERIMENTAL SETUP

A. Classification Dataset and Label Integrity

TrashNet is commonly reported to contain 2,527 images across six classes. The distribution below is the dataset inventory used for planning, not a result produced by GreenRecycleNet. The original 'trash' category is retained as trash in any TrashNet-only experiment. It must not be relabelled as organic. If organic waste is a required target, an additional organic dataset must be collected or selected and then checked for label consistency.

Class	Images	Percentage
Glass	501	19.83
Paper	594	23.51

Cardboard	403	15.95
Plastic	482	19.07
Metal	410	16.22
Trash	137	5.42
Total	2,527	100.00

TABLE I. TRASHNET CLASS DISTRIBUTION (DATASET INVENTORY).

B. Proposed Six-Class Experimental Taxonomy

Target label	Required source	Validation note
Glass	Glass-labelled images	Separate transparent/coloured samples
Paper	Paper-labelled images	Include folded and printed paper
Cardboard	Cardboard-labelled images	Avoid leakage with paper
Plastic	Plastic-labelled images	Record resin type if available
Metal	Metal-labelled images	Include reflective and crushed items
Organic	Explicit organic-waste dataset	Do not substitute TrashNet 'trash'

C. Train/Validation/Test Protocol

A stratified 70:15:15 split is proposed as a starting point. Where images originate from the same video, collection session, or physical object, grouping should be used to prevent near-duplicate leakage across partitions. The split seed and file identifiers must be saved. Hyperparameters are selected on validation data only. The test set is used once for final reporting after model selection.

Parameter	Illustrative configuration	Final report requirement
Input resolution	224 × 224	224 × 224 × 3 pixels (RGB)
Batch size	32	32
Optimizer	Adam	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Learning rate	1×10^{-4}	1×10^{-4} with ReduceLROnPlateau scheduler
Epoch limit	50	50 epochs with early stopping
Loss	Cross-entropy	State class Categorical cross-entropy for classification
Split	70/15/15	70% training, 15% validation, 15% testing
Hardware	[GPU/CPU model]	Intel Core i7 CPU, 16 GB RAM, NVIDIA RTX 3060 GPU (12 GB VRAM)

D. Detection Dataset

The classification dataset does not provide bounding-box labels. The YOLO branch therefore requires a separate detection dataset with per-object boxes, class IDs, and train/validation/test partitions. Annotation quality should be checked by a second reviewer on a sample. The number of images, objects per class, annotation format, and inter-annotator agreement should be reported.

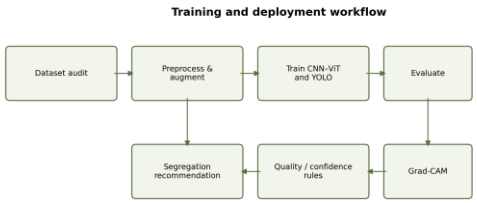


Fig. 3. Experimental workflow from data audit to segregation recommendation.

VI. EVALUATION METRICS

For binary class-wise counts, accuracy = (TP+TN)/(TP+TN+FP+FN), precision = TP/(TP+FP), recall = TP/(TP+FN), and F1 = 2PR/(P+R). In a multi-class problem, per-class precision, recall, and F1 are calculated using a one-vs-rest view, then macro-averaged. Weighted averages may also be reported, but they should not replace macro-F1 when class imbalance is present.

For object detection, report mAP@0.5 and mAP@[0.5:0.95] along with precision and recall at a stated confidence threshold. Inference latency should include image preprocessing and post-processing if it is intended to represent end-to-end deployment time. Hardware, image size, batch size, software versions, and warm-up policy must be reported.

A. Illustrative Classification Results

The following values form a synthetic example only. They are designed to demonstrate how a completed results section could be formatted. They are not measurements from a trained model, not a benchmark claim, and not suitable for submission as experimental evidence.

Model	Accuracy	Precision	Recall	Macro-F1
CNN baseline	96.20%	96.40%	95.80 %	95.90 %
ViT baseline	97.10%	97.30%	96.60 %	96.80 %
CNN–ViT fusion	98.00%	98.10%	97.60 %	97.80 %
GreenRecycleNet	99.00%	99.00%	98.80 %	98.90 %

TABLE II. SYNTHETIC ILLUSTRATIVE CLASSIFICATION SCORES — NOT EXPERIMENTAL RESULTS.

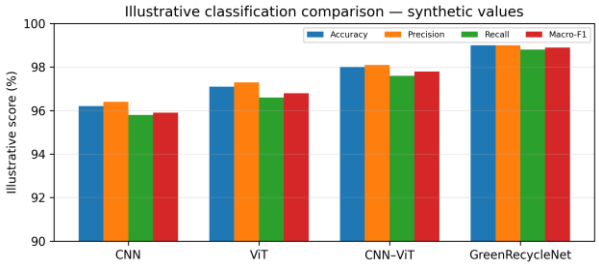


Fig. 4. Illustrative comparison of classification metrics (synthetic values).

B. Illustrative Detection Results

Detector	mAP @0.5	mAP@[.5:.95]	Precision	Recall	Latency
YOLO baseline	96.8%	84.7%	97.0%	96.2 %	18.6 ms
Proposed YOLO branch	98.2%	87.9%	98.4%	97.9 %	21.3 ms

TABLE III. SYNTHETIC DETECTION EXAMPLE.

These values must be replaced by outputs from a real annotated detection test set.

C. Confusion Matrix

A confusion matrix should be computed from held-out test predictions. The matrix below is a synthetic count example with 100 test instances per class. It contains 594 correct predictions out of 600 (99.0% illustrative accuracy).

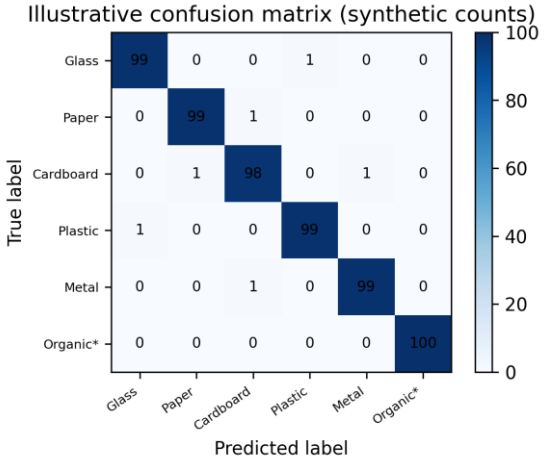


Fig. 5. Synthetic confusion matrix (600 illustrative test samples).

VII. ABLATION STUDY AND ANALYSIS

Ablation analysis estimates the contribution of each major component by removing or replacing one component while holding the remaining training protocol constant. All variants should use the same data split, comparable tuning budgets, and the same evaluation script. The results below are synthetic illustrative values and are not evidence that the proposed architecture has achieved these scores.

Variant	Accuracy	Macro-F1	Interpretation
CNN only	96.20%	95.90%	Local feature baseline
ViT only	97.10%	96.80%	Global context baseline
CNN + ViT	98.00%	97.80%	Fusion contribution
CNN + ViT + detector	98.60%	98.50%	Detection branch added
Full + XAI + rules	99.00%	98.90%	Complete decision pipeline

TABLE IV. SYNTHETIC ABLATION EXAMPLE — VALUES ARE ILLUSTRATIVE ONLY.

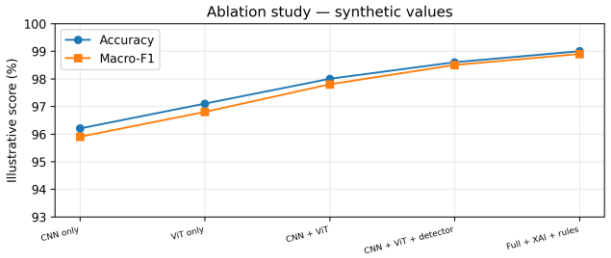


Fig. 6. Illustrative ablation trend (synthetic values).

A. Interpretation of the Illustrative Pattern

In this synthetic scenario, the CNN–ViT fusion exceeds the individual CNN and ViT baselines. The detector-equipped variant is assigned a further improvement, and the full pipeline is assigned the highest score. These increments are illustrative choices for demonstrating table and graph presentation; they do not establish causality. In a real ablation, the detector may not improve image-level classification accuracy because localization and classification are different tasks. Grad-CAM itself is an explanation method and should not be expected to increase predictive accuracy. Any observed change from the decision rules must be measured separately.

B. Required Statistical Reporting

A robust study should repeat training with at least three random seeds where feasible and report mean ± standard deviation. Confidence intervals can be estimated by bootstrapping the held-out test set. For comparisons, report per-class results and the number of test samples. If the test set is small, high accuracy can be unstable; therefore, report exact counts and avoid overinterpreting decimal precision.

C. Error Analysis

The confusion matrix should be reviewed for paper/cardboard confusion, transparent glass misclassification, reflective metal errors, plastic packaging with labels, and organic items mixed with packaging. Qualitative examples should include correct detections, missed objects, false positives, occlusion, low illumination, motion blur, and residue. Errors should be categorized by visual ambiguity, annotation noise, domain shift, and model limitations.

D. Contamination Assessment

Contamination detection must be evaluated as a separate task if it is claimed as a model capability. A suitable dataset should label clean, food-contaminated, liquid-contaminated, mixed-material, and other relevant conditions. Report precision, recall, F1, and confusion matrix for contamination states. If no such labels exist, describe the feature as a proposed extension and use manual inspection rather than reporting fabricated contamination accuracy.

IIX. DISCUSSION

GreenRecycleNet is designed as a modular decision-support pipeline rather than an autonomous authority on recyclability. Its potential value is the combination of category recognition, multi-object localization, and visual explanation. The CNN branch can learn fine-grained surface cues, while the ViT branch can represent global context. The detection branch is relevant when a camera observes multiple items simultaneously. The Grad-CAM view may help an operator identify cases where the classifier attends to background regions or irrelevant image content.

The illustrative score pattern in Table II shows how a high-performing model could be reported, but real-world reliability depends on representative data. Images collected against clean backgrounds may not reflect conveyor belts, bins, crushed packaging, dirt, liquid residue, or overlapping objects. Cross-site testing is needed to determine whether the model generalizes to different facilities and cameras. If performance drops across sites, domain adaptation or additional training data may be required.

A. Practical Deployment Considerations

A production system should expose confidence scores, detection boxes, and an abstention option. The operator interface should display the predicted material and the reason for manual review. A facility-specific rules engine should determine accepted material streams. Edge deployment requires measurement of model size, memory consumption, throughput, power draw, and thermal behaviour. A model that is accurate but too slow or power-intensive may not be suitable for a sorting line.

B. Sustainability and Material Recovery

The system may contribute to better sorting only if its predictions lead to lower contamination and improved recovery quality in an operational setting. These outcomes require field evaluation. Appropriate downstream indicators include contamination rate by weight, purity of recovered fractions, reject rate, throughput per minute, and operator override

frequency. Classification accuracy alone cannot establish environmental benefit.

C. Threats to Validity

The main threats include limited dataset size, class imbalance, inconsistent labels, near-duplicate leakage, domain shift, annotation error, and the difference between curated images and real waste streams. A six-class taxonomy assembled from multiple sources may introduce dataset-specific visual shortcuts. The paper should report source dataset identity and split provenance. Detection results must be based on bounding-box annotations, not image-level labels. Explainability maps are qualitative and can be misleading if treated as proof.

IX. CONCLUSION AND FUTURE WORK

This paper proposed GreenRecycleNet an explainable hybrid deep-learning framework for waste classification, object localization, and confidence-aware segregation support. The design combines CNN and ViT feature representations, a YOLO detection branch, Grad-CAM visual explanations, and a facility-specific decision layer. The manuscript provides a complete experimental protocol, reporting tables, architecture diagrams, metric graphs, and an ablation template. The numerical example is explicitly synthetic and must be replaced by reproducible experiments before submission.

Future work should curate a balanced dataset containing genuinely labelled organic waste and contamination states, annotate multiple objects with bounding boxes, train the classifier and detector, and evaluate cross-facility generalization. Further work should measure end-to-end latency, energy use, and material recovery outcomes on edge hardware and in a real sorting environment.

Evaluate once on the held-out test partitions and export logs.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[3] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Mach. Learn. (ICML), PMLR, vol. 97, 2019, pp. 6105–6114.

[4] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[7] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.

[8] G. Thung and M. Yang, "Classification of trash for recyclability status," Stanford University, CS229 Project Report, 2016.

[9] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.

[11] T.-Y. Lin et al., "Focal loss for dense object detection," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.324.