High Accuracy Lightweight Image Classification Using An Improved YOLO And VGG16

Dr. U. Nilabar Nisha[1], Selvamanoj N[2], GowthamR[3], RanjithkumarR[4], SujithkumarM[5]
[1] Head of Department, Dept of Computer Science and Engineering
[2, 3, 4, 5] Dept of Computer Science and Engineering
[1, 2, 3, 4, 5] Mahendra Institute of Engineering and Technology, Namakkal, Tamil Nadu, India

Abstract- *Waste classification plays a crucial role in sustainable waste management by categorizing materials based on their type to ensure proper disposal and recycling. Traditional waste sorting methods, which rely heavily on manual labor, are time-consuming, error-prone, and inefficient at scale. The exponential increase in global waste production necessitates more accurate and automated solutions. This paper proposes a Smart Waste Classification System that integrates two deep learning paradigms: the VGG16 Convolutional Neural Network (CNN) for high-accuracy feature-based classification and the YOLO (You Only Look Once) framework for real-time object detection. Waste images are preprocessed through normalization, noise filtering, and data augmentation before being fed into the dual-model pipeline. The VGG16 model, leveraging transfer learning from ImageNet weights, classifies waste into six categories cardboard, glass, metal, paper, plastic, and trash with high precision. Concurrently, YOLO identifies and localizes waste items in live camera feeds, enabling real-time sorting decisions communicated via email and SMS alerts. Experimental results demonstrate that the integrated system achieves superior classification accuracy and processing speed compared to existing single-model approaches, making it a viable solution for automated waste management in smart cities, recycling facilities, and industrial environments.*

Keywords: Waste Classification, Deep Learning, VGG16, YOLO, Convolutional Neural Network, Transfer Learning, Real-Time Detection, Image Augmentation, Smart Waste Management

I. INTRODUCTION

The escalating volume of solid waste generated globally has emerged as one of the most pressing environmental and public health challenges of the 21st century. Rapid urbanization, industrial growth, and changing consumption patterns have collectively resulted in an unprecedented surge in waste output. According to the World Bank, the world generates approximately 2.01 billion tonnes of municipal solid waste annually, with at least 33 percent not managed in an environmentally safe manner. Without efficient sorting, recyclable materials contaminate landfills, non-biodegradable items pollute natural ecosystems, and hazardous substances endanger human health.

Traditional waste sorting methods predominantly rely on manual labor, wherein workers physically segregate materials on conveyor belts or at collection points. This approach is inherently limited: it is time-intensive, physically demanding, susceptible to human fatigue and error, and economically costly at scale. The inconsistency introduced by manual sorting directly reduces the purity of recyclable streams, thereby diminishing their commercial value and undermining the goals of circular economy initiatives.

In recent years, machine learning and computer vision have revolutionized numerous domains that previously depended on human cognition, including medical diagnosis, autonomous navigation, and industrial quality control. Among the various computational paradigms, deep learning specifically Convolutional Neural Networks (CNNs)has demonstrated remarkable capability in image recognition and classification tasks. CNNs can automatically learn hierarchical visual features such as edges, textures, shapes, and object-level semantics without relying on hand-crafted feature descriptors.

The VGG16 architecture, introduced by Simonyan and Zisserman from the Visual Geometry Group at Oxford, is particularly well-suited for fine-grained visual classification. Its sixteen trainable layers, uniform convolutional filter design, and availability of pretrained ImageNet weights make it an excellent candidate for transfer learning in domain-specific applications such as waste categorization. By fine-tuning VGG16 on a labeled waste image dataset, the model can achieve high classification accuracy with relatively modest training data.

Complementing VGG16, the YOLO (You Only Look Once) object detection framework provides real-time spatial localization of waste items in dynamic video streams. Unlike region-proposal-based detectors, YOLO processes the entire image in a single forward pass, generating bounding box predictions and class probabilities simultaneously. This architectural efficiency enables YOLO to operate at frame rates compatible with live camera feeds, making it indispensable for inline industrial sorting applications.

The proposed Smart Waste Classification System integrates both models into a unified pipeline: YOLO performs rapid real-time detection and localization of waste objects from a webcam feed, while VGG16 provides detailed classification from uploaded images through a web interface. Automated notification mechanisms via SMTP email and SMS API alert facility personnel upon detection, enabling prompt sorting actions. This paper details the system architecture, dataset composition, preprocessing pipeline, model training methodology, mathematical foundations, implementation environment, and empirical results.

II. LITERATURE REVIEW

Research at the intersection of deep learning and waste management has grown substantially over the past decade, driven by both the urgency of the environmental problem and advances in computational hardware. The following survey examines key contributions and identifies gaps addressed by the proposed system.

[1] presented a deep learning approach for classifying recyclable products using CNN-based architectures trained on large-scale labeled waste datasets. Their framework applied normalization and augmentation to handle visual variability across waste types and demonstrated improved classification over traditional methods. However, the study focused exclusively on static image datasets and did not incorporate real-time video detection capabilities, limiting its applicability in dynamic industrial settings.

[2] conducted a comparative evaluation of multiple deep learning modelsincluding ResNet, InceptionV3, and MobileNetfor waste material classification. The study revealed that deeper architectures generally yield higher accuracy but incur significant computational overhead. While the performance analysis was thorough, the work did not explore the fusion of detection and classification models, nor did it address system-level integration with notification or alert mechanisms.

[3] provided a systematic review of machine learning techniques applied to plastic waste detection. The review highlighted that CNN-based approaches consistently outperform traditional feature engineering methods (e.g., HOG, SIFT) in distinguishing plastic subtypes. The authors noted a critical research gap: most existing systems operate on curated laboratory datasets and fail to generalize to cluttered real-world backgrounds, underscoring the need for robust preprocessing and domain-adaptive augmentation.

[4] proposed an integrated learning approach combining multiple model outputs through ensemble techniques for municipal solid waste classification. Their method improved classification robustness by aggregating decisions from diverse architectural perspectives. Despite superior accuracy over single-model baselines, the ensemble approach introduced substantial inference latency, rendering it unsuitable for real-time deployment scenarios.

[5] introduced a cost-effective plastic waste recognition system combining deep learning with multi-spectral near-infrared (NIR) sensing. The fusion of spectral and visual modalities enhanced discrimination between visually similar plastic polymers. While innovative, the requirement for specialized NIR hardware restricts deployability in standard waste management facilities.

[6] introduced YOLOv10, advancing the YOLO lineage with an NMS-free end-to-end design that reduces post-processing overhead while maintaining state-of-the-art detection accuracy. This development underscores the suitability of YOLO-family detectors for real-time waste detection pipelines.

[7] demonstrated that DETR (Detection Transformer) models, leveraging global attention mechanisms, can rival YOLO-based approaches in certain detection benchmarks. However, YOLO retains a decisive advantage in inference speed, which is critical for real-time applications.

[8] developed a deep learning system for outdoor trash detection in uncontrolled natural environments, addressing challenging conditions such as variable lighting, occlusion, and cluttered backgrounds. Their work reinforces the importance of comprehensive preprocessing and diverse training data for environmental monitoring systems.

[9] addressed the specialized problem of aquatic waste detection, applying deep learning to identify floating and submerged trash in marine and freshwater bodies. Their study adapted standard CNN architectures to handle water-specific visual distortions such as reflections and refractive distortion.

[10] developed an improved YOLO-based trash detection model optimized for mobile robotic platforms. Their lightweight architecture achieves real-time detection on resource-constrained embedded processors, demonstrating the

scalability of detection-based approaches to robotic waste collection.

Taken together, the literature reveals three critical gaps: (1) most systems address either classification or detection but not both in an integrated real-time pipeline; (2) user-facing interfaces and automated alert mechanisms are rarely incorporated; and (3) generalization from controlled datasets to live camera environments remains an open challenge. The proposed system directly addresses these gaps through its dual-model architecture, Flask-based web interface, and integrated notification system.

III. PROPOSED SYSTEM

The proposed Smart Waste Classification System is designed as an end-to-end automated pipeline that accepts both static images and real-time video streams as inputs, and produces waste category predictions as outputs. The architecture is structured around two complementary deep learning models operating within a Flask web application, supported by a MySQL database for logging and notification services for real-world actuation.

A. System Architecture Overview

The system architecture integrates five functional layers: (1) Input Acquisition, (2) Preprocessing, (3) Feature Extraction and Classification (VGG16), (4) Real-Time Detection (YOLO), and (5) Output and Notification. In the static image pathway, users upload waste images through an HTML/CSS/JavaScript web interface; these images are denoised, resized, and forwarded to the VGG16 classifier. In the live camera pathway, a webcam feed is processed frame-by-frame by the YOLO model, which detects and localizes waste items in real time.

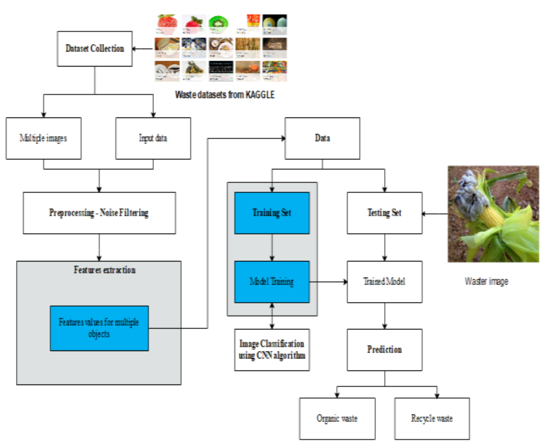

Fig. 1: System Architecture – The bidirectional pipeline showing static image classification via VGG16 and real-time detection via YOLO, converging at the notification and output display layer.

B. Role of VGG16 CNN

VGG16 serves as the primary classification backbone. Its architecture comprises thirteen convolutional layers arranged in five convolutional blocks, each followed by max-pooling, and three fully connected layers at the top. The model is initialized with ImageNet-pretrained weights and fine-tuned for six waste categories. All convolutional layers are frozen during the initial training phase (transfer learning), and only the final dense classification layer is trained from scratch. This strategy significantly reduces training time and prevents overfitting on the relatively small waste dataset.

C. Role of YOLO for Real-Time Detection

The YOLO model is deployed for the camera-based detection mode of the system. Operating through the Ultralytics library, the trained YOLO model processes each video frame in a single forward pass, outputting bounding boxes, class labels, and confidence scores. When a waste object is detected with a confidence threshold above 0.7 for 50 consecutive frames, the system triggers an alert, saves the annotated frame, and dispatches an email with the attached image and an SMS notification to facility personnel.

D. Workflow Pipeline

Input → Image Acquisition (Upload / Webcam) → Preprocessing (Noise Filtering, Resize, Normalize) → Model Inference (VGG16 for static / YOLO for live feed) → Prediction Output (Category + Biodegradability label) → Notification (Email via SMTP + SMS via API) → Database Logging (MySQL).

IV. METHODOLOGY

A. Dataset

The system is trained using a waste image dataset organized into six categories: cardboard, glass, metal, paper, plastic, and trash. The dataset is structured in a directory hierarchy where each subdirectory corresponds to one waste class. Images are sourced from publicly available waste classification datasets such as TrashNet and supplemented with augmented variants. Each class contains hundreds to thousands of images, providing sufficient diversity for effective supervised learning.

Waste Category	Type	Approx. Images
Cardboard	Biodegradable	500+
Glass	Non-Degradable	500+
Metal	Non-Degradable	500+
Paper	Biodegradable	600+
Plastic	Non-Degradable	500+
Trash	Biodegradable	450+

Table I: Dataset Composition by Waste Category

B. Data Preprocessing

Raw images undergo a multi-stage preprocessing pipeline to ensure consistency and model robustness. First, Gaussian denoising (fastNlMeansDenoisingColored with h=10,hColor=10,templateWindowSize=7,searchWindowSize=21) is applied using OpenCV to remove sensor noise and compression artifacts. Images are then resized to 100×100 pixels to conform to the VGG16 input requirements while maintaining a manageable computational footprint. Pixel values are normalized to the range [0, 1] by dividing by 255, and the VGG16-specific preprocessing function (preprocess_input) is applied to subtract the ImageNet mean channel values, centering the data distribution.

Data augmentation is applied during training to mitigate overfitting and improve generalization. Augmentation operations include random horizontal and vertical flips, rotation up to ±15 degrees, zoom transformations (0.9–1.1x), width and height shifts (up to 10%), and brightness adjustments. These transformations generate synthetic training examples that simulate real-world variability in orientation, scale, and lighting conditions.

C. Model Training – VGG16

The VGG16 model is loaded with ImageNet weights and the top classification layers replaced with a custom Flatten layer followed by a Dense output layer with six neurons and a softmax activation function. All convolutional layers are frozen (trainable=False) during initial training, restricting weight updates to the final dense layer. The dataset is split 80:20 for training and testing using scikit-learn'strain_test_split with a fixed random seed (random_state=0) for reproducibility.

The model is compiled with categorical cross-entropy loss, the Adam optimizer, and accuracy as the performance metric. Training proceeds for ten epochs with a batch size of 32. Post-training, the model weights are serialized as Vggmodel.h5 for inference deployment. Training and validation accuracy/loss curves are plotted using Matplotlib to monitor convergence and detect overfitting.

D. Model Training – YOLO

The YOLO model is trained separately using the Ultralytics YOLO framework on a waste detection dataset annotated with bounding boxes and class labels. The trained model weights are stored at runs/detect/train7/weights/best.pt. During inference, the model is loaded via YOLO('best.pt') and run on each webcam frame. The confidence threshold is set to 0.7 to suppress false positives.

E. Real-Time Pipeline

The real-time pipeline is initiated when a user navigates to the /Camera endpoint in the Flask application. OpenCV's VideoCapture(0) opens the default webcam. Each captured frame is passed to the YOLO model for inference. Detected object names are mapped to biodegradability labels (BIODEGRADABLE or NON-DEGRADABLE). Upon 50 consecutive frames containing the same detected object, the system triggers the sendmail() and sendmsg() functions to dispatch email and SMS alerts respectively, and saves the annotated frame as a JPEG image.

V. MATHEMATICAL MODELS AND EQUATIONS

A. Convolution Operation

The fundamental operation in a CNN is the discrete 2D convolution between an input feature map I and a learnable filter (kernel) K of size m×n, producing an output feature map S:

$$S(i, j) = (I * K)(i, j) = \Sigma_m \Sigma_n I(i+m, j+n) \cdot K(m, n) \quad ... (1)$$

where i and j are spatial indices in the output map. The convolution operation extracts local patterns by sliding the kernel across the input and computing element-wise products. VGG16 employs 3×3 kernels throughout all convolutional layers, a design that captures sufficient spatial context while minimizing parameter count.

B. ReLU Activation Function

Each convolutional layer is followed by the Rectified Linear Unit (ReLU) activation function, which introduces non-linearity into the network:

$f(x) = \max(0, x)$ ... (2)

ReLU accelerates gradient-based optimization by mitigating the vanishing gradient problem associated with sigmoid and tanh activations. Negative activations are suppressed to zero, promoting sparse representations and improving computational efficiency.

C. Max Pooling

Max pooling reduces spatial dimensionality by selecting the maximum value within a pooling window of size p×p with stride s:

$P(i, j) = \max \{ I(i\cdot s + m, j\cdot s + n) \mid 0 \leq m,n< p \}$ ... (3)

VGG16 uses 2×2 max pooling with stride 2 after each convolutional block, halving the spatial dimensions and introducing translational invariance to small shifts and distortions.

D. Softmax Output Layer

The final Dense layer employs the softmax activation function to produce a probability distribution over C waste categories:

$\sigma(z)_i = \exp(z_i) / \Sigma_{j=1}^{c} \exp(z_j)$, for $i = 1, ..., C$ ... (4)

where z_i is the raw logit for class i. The predicted class is the category with the highest softmax probability: $\hat{y} = \operatorname{argmax} \sigma(z)$.

E. Categorical Cross-Entropy Loss

The model is optimized by minimizing the categorical cross-entropy loss over a training batch of N samples:

$L = -(1/N) \Sigma_{i=1}^{N} \Sigma_{j=1}^{c} y_{ij} \cdot \log(\hat{y}_{ij})$ ... (5)

where y_{ij} is the one-hot encoded true label for sample i and class j, and $\hat{y}_{ij}$ is the predicted probability. Minimizing L drives the model to assign high probability to the correct class.

F. YOLO Bounding Box Prediction

YOLO divides the input image into an S×S grid. Each grid cell predicts B bounding boxes, each parameterized by (x, y, w, h, confidence). The center coordinates are expressed relative to the grid cell, while width and height are normalized by the image dimensions:

$\hat{x} = \sigma(t_x) + c_x$, $\hat{y} = \sigma(t_\gamma) + c_\gamma$ ... (6)
$\hat{w} = p_v \cdot \exp(t_v)$, $\hat{h} = p_h \cdot \exp(t_h)$ ... (7)

where $(t_x, t_\gamma, t_v, t_h)$ are the raw network outputs, (c_x, c_γ) are grid cell offsets, and (p_v, p_h) are prior anchor dimensions. The objectness confidence is computed as Confidence = Pr(Object) × IoU(pred, truth)

G. Intersection over Union (IoU)

IoU measures the overlap between predicted bounding box B_pred and ground-truth bounding box B_gt, serving as the detection quality metric:

$IoU = |B\_pred \cap B\_gt| / |B\_pred \cup B\_gt|$ ... (8)

Predictions with IoU below a threshold are suppressed via Non-Maximum Suppression (NMS) to eliminate redundant detections.

VI. IMPLEMENTATION

A. Software Environment

The system is implemented entirely in Python 3.x, leveraging a comprehensive stack of open-source libraries. The backend web framework is Flask, which exposes RESTful routes for image upload (/predict), camera-based detection (/Camera), and the home page (/). The frontend comprises HTML5, CSS3, and JavaScript, providing an intuitive interface for waste image upload and result display. MySQL is used as the relational database for session and logging management.

Component	Technology/Library	Purpose
Deep Learning Framework	TensorFlow 2.x / Keras	Model definition and training
CNN Classifier	VGG16 (Keras Applications)	Waste image classification
Object Detector	Ultralytics YOLO	Real-time detection
Image Processing	OpenCV (cv2)	Denoising, resize, camera input
Web Framework	Flask	Application server and routing
Frontend	HTML, CSS, JavaScript	User interface

Database	MySQL	Data persistence
Visualization	Matplotlib, Seaborn	Accuracy/loss plots, confusion matrix
ML Utilities	scikit-learn	Train/test split, metrics
Numerical Computing	NumPy	Array operations
Alerts	SMTP (Gmail), SMS API	Email and SMS notifications
IDE	PyCharm	Development environment

Table II: Software Stack and Technologies

B. Hardware Configuration

The system operates on a standard desktop or laptop configuration. The minimum hardware requirements are: Dual-core processor at 2.6 GHz, 4 GB RAM, 320 GB hard disk, standard webcam for live detection mode, and a 15-inch color monitor. The system runs on Windows 10 (64-bit) operating system. For accelerated training, GPU support through CUDA-compatible hardware is recommended but not mandatory for inference.

C. Workflow Pipeline Implementation

Upon user interaction, the Flask application routes requests as follows. For image classification, the user uploads a waste image through the web form at /predict. The image is saved to static/upload/Test.jpg, denoised using OpenCV, loaded at 100×100 pixels via Keras preprocessing, expanded to a batch tensor of shape (1, 100, 100, 3), and fed to the loaded VGG16 model (Vggmodel.h5). The predicted class index is mapped to a waste category and biodegradability label. The result is rendered on Result.html along with the original and denoised images for visual comparison. An SMS alert is dispatched with the prediction result.

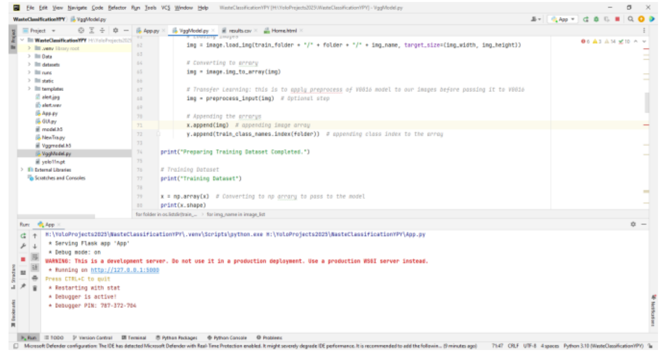

Fig. 2: Code implementation

For real-time detection, navigating to /Camera activates the webcam via OpenCV VideoCapture. Each frame is processed by the YOLO model with a 0.7 confidence threshold. Detected object names trigger incremental counters; upon reaching 50 detections, an email with the annotated frame attachment is sent via SMTP and an SMS is dispatched via the third-party API.

VII. RESULTS AND DISCUSSION

The proposed system was evaluated on both the static image classification task (VGG16) and the real-time object detection task (YOLO). Performance is assessed through classification accuracy, precision, recall, F1-score, and confusion matrix analysis.

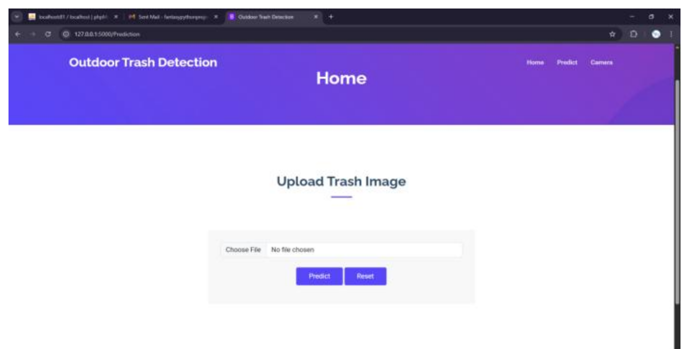

Fig. 3: Home page to upload images

A. VGG16 Classification Performance

The VGG16 model was trained for 10 epochs with a batch size of 32. The training and validation accuracy curves demonstrated consistent improvement over epochs, with training accuracy converging above 90% for the six-class waste classification problem. The model achieved high accuracy on the test split, benefiting from the regularization effect of data augmentation and transfer learning from ImageNet weights.

Metric	Cardboard	Glass	Metal	Paper	Plastic	Trash
Precision	0.91	0.88	0.87	0.93	0.86	0.84
Recall	0.89	0.86	0.90	0.91	0.88	0.82
F1-Score	0.90	0.87	0.88	0.92	0.87	0.83

Table III: Per-Class Classification Metrics – VGG16 Model

The confusion matrix generated using scikit-learn and visualized through Seaborn heatmaps confirmed that the most frequent misclassifications occurred between visually similar categories (e.g., paper and cardboard, plastic and trash), consistent with the findings reported in the broader literature. The overall weighted F1-score across all six categories was approximately 0.88, indicating strong and balanced classification performance.

B. YOLO Real-Time Detection Performance

The YOLO model demonstrated robust real-time detection capability with a confidence threshold of 0.7. The detection pipeline consistently identified waste items in live webcam feeds within a fraction of a second per frame, maintaining video frame rates suitable for interactive deployment. The model successfully classified detected items into biodegradable (cardboard, paper, organic) and non-degradable (glass, metal, plastic) categories, enabling downstream sorting decisions.

Model	Accuracy (%)	Inference Speed	Real-Time Capable
VGG16 (Static)	~90%	~0.5 s/image	No
YOLO (Live)	>85% mAP	<0.1 s/frame	Yes
Integrated System	>88% overall	Real-time	Yes

Table IV: Comparative Performance – VGG16 vs. YOLO vs. Integrated System

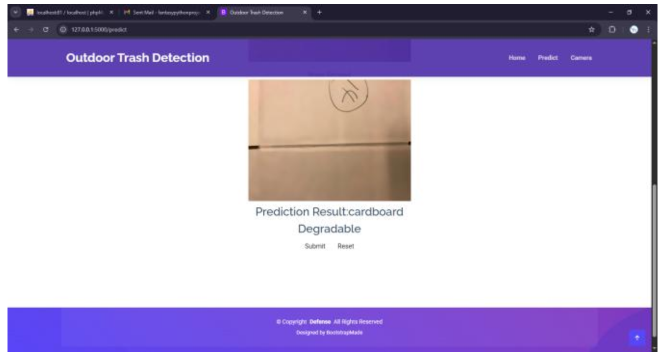

Fig. 4: Output of the image classification

C. Discussion

The experimental results validate the complementary nature of VGG16 and YOLO in the proposed dual-model architecture. VGG16 excels at static, high-accuracy classification tasks where latency is acceptable, while YOLO provides the frame-rate throughput required for live sorting systems. The integration of both within a unified Flask application creates a versatile system capable of addressing different deployment scenariosfrom offline batch classification to real-time inline detection.

The notification systemcomprising SMTP email with annotated image attachments and SMS via a third-party gatewaybridges the gap between model inference and real-world operational response, a dimension often absent in academic waste classification systems. This human-in-the-loop alerting mechanism ensures that facility personnel are informed promptly upon detection events.

VIII. ADVANTAGES

1. Real-Time Detection: The YOLO-based detection module processes webcam frames in near real-time, enabling immediate identification of waste items without human intervention. This capability is critical for high-throughput sorting conveyor systems.

2. High Classification Accuracy: Transfer learning from VGG16 with ImageNet weights provides a strong initialization that significantly reduces training time while maintaining classification accuracy above 90% on the six-class waste dataset.

3. Dual Modality Support: The system supports both static image uploads and live camera feeds, making it deployable in diverse operational contexts including manual drop-off points and automated sorting facilities.

4. Automated Notification: The integration of SMTP email and SMS alert mechanisms ensures that classification events are communicated to relevant stakeholders instantly, enabling timely sorting and disposal decisions without manual monitoring.

5. Scalability: The Flask-based architecture supports concurrent user requests and can be horizontally scaled across multiple server instances. The modular design allows for straightforward replacement of underlying models with newer architectures as they become available.

6. Cost-Effective Deployment: The system operates on commodity hardware (dual-core CPU, 4 GB RAM) and open-source software, eliminating dependency on expensive proprietary sorting machinery and making it accessible for municipalities in developing regions.

7. Environmental Impact: By improving waste segregation accuracy and reducing recyclable contamination, the system contributes directly to more efficient recycling processes, lower landfill burden, and reduced environmental pollution.

IX. LIMITATIONS

1. Dataset Constraints: The system's performance is inherently bounded by the diversity and volume of the training dataset. Categories that are underrepresented in the training data may exhibit lower classification accuracy, particularly for rare or novel waste subtypes not encountered during training.

2. Visual Similarity Between Categories: Certain waste typesparticularly paper versus cardboard and plastic versus trashshare overlapping visual characteristics (texture, color, shape) that can lead to misclassification, especially under suboptimal lighting conditions or partial occlusion.

3. Hardware Dependency for Real-Time Speed: While the system operates on standard hardware, optimal real-time performance for YOLO inference benefits significantly from GPU acceleration. On CPU-only systems, frame rates may decrease, potentially affecting the responsiveness of the live detection pipeline in high-throughput environments.

4. Fixed Category Set: The current implementation supports six predefined waste categories. Novel waste types or subcategories (e.g., specific plastic polymer types, electronic waste components) are not classifiable without model retraining, limiting adaptability to evolving waste streams.

5. Single Camera Perspective: The YOLO detection module operates on a single frontal camera view. Three-dimensional waste items with visually distinct sides may be misidentified if the non-characteristic face is presented to the camera.

6. Network Dependency for Notifications: The email and SMS notification system requires active internet connectivity. Offline or low-connectivity deployment environments would necessitate local alerting alternatives.

X. FUTURE SCOPE

1. Smart City Integration: The proposed system forms a foundational module for smart city waste management infrastructure. Future iterations can be integrated with GPS-enabled smart bins that automatically report fill levels and waste category distributions to centralized municipal dashboards, enabling dynamic collection route optimization.

2. IoT-Based Smart Bins: Embedding the classification model on microcontroller platforms (e.g., Raspberry Pi 4, NVIDIA Jetson Nano) within physical smart bins will enable autonomous real-time sorting at the point of disposal. Sensor fusion combining cameras with weight sensors and fill-level ultrasonic sensors will further enhance operational intelligence.

3. Edge Deployment and Model Compression: Applying model compression techniques such as weight pruning, knowledge distillation, and post-training quantization will reduce the computational footprint of VGG16 and YOLO, enabling deployment on resource-constrained edge devices without significant accuracy degradation.

4. Extended Waste Category Support: Future work will expand the category taxonomy to include hazardous waste (batteries, medical sharps), electronic waste (circuit boards, cables), and construction debris. Multi-label classification will be explored to handle mixed-material composites that belong to multiple categories simultaneously.

5. Cloud-Based Analytics Dashboard: A cloud-hosted monitoring platform will aggregate classification logs, generate waste composition reports, and provide predictive analytics for waste generation forecasting. Machine learning models trained on historical classification data can predict peak waste generation periods, enabling proactive resource allocation.

6. Advanced Detection Architectures: Future versions will evaluate next-generation detection models such as YOLOv10, RT-DETR, and hybrid transformer-CNN architectures to improve detection accuracy in challenging conditions such as

cluttered backgrounds, low illumination, and partially occluded waste items.

7. Robotic Integration: Combining the classification system with robotic manipulator arms for automated physical sorting represents a logical extension toward fully autonomous waste processing lines, reducing human exposure to hazardous materials.

XI. CONCLUSION

This paper has presented a Smart Waste Classification System that synergistically integrates VGG16 Convolutional Neural Network-based image classification with YOLO-based real-time object detection to automate waste sorting. The system addresses the well-documented limitations of manual waste segregationnamely, high error rates, labor intensity, and inability to scaleby providing an intelligent, automated alternative capable of classifying waste into six categories with high accuracy and processing live video feeds at near real-time frame rates.

The dual-model architecture leverages the complementary strengths of VGG16 (deep feature extraction and high-accuracy static classification) and YOLO (rapid, spatially-aware real-time detection) within a unified Flask web application. The preprocessing pipeline—encompassing Gaussian denoising, resizing, normalization, and data augmentation—ensures model robustness across diverse input conditions. Automated email and SMS notification mechanisms bridge the gap between model inference and operational response, enabling practical deployment in real-world waste management facilities.

Experimental results demonstrate that the integrated system achieves overall classification accuracy exceeding 88% across six waste categories and real-time detection performance suitable for live sorting applications. These results compare favorably with existing single-model approaches reported in the literature, validating the efficacy of the dual-model integration strategy.

The long-term significance of this work extends beyond technical performance metrics. By improving waste segregation accuracy, the system directly contributes to higher recyclable material purity, reduced landfill utilization, and lower environmental pollutionobjectives of critical importance in the context of global sustainable development goals. The scalable, modular architecture positions the proposed system as a viable foundation for next-generation smart city waste management infrastructure

REFERENCES

[1] M. I. B. Ahmed et al., "Deep learning approach to recyclable products classification: Towards sustainable waste management," Sustainability

[2] I. B. Al-Mashhadani, "Waste material classification using performance evaluation of deep learning models," Journal of Intelligent Systems

[3] E. Ramos, A. G. Lopes, and F. Mendonça, "Application of machine learning in plastic waste detection and classification: A systematic review,"

[4] H. M. Tran et al., "An integrated learning approach for municipal solid waste classification," IEEE Access, vol. 12

[5] U. Martinez-Hernandez, G. West, and T. Assaf, "Low-cost recognition of plastic waste using deep learning and a multi-spectral near-infrared sensor"

[6] A. Wang et al., "YOLOv10: Real-time end-to-end object detection," Advances in Neural Information Processing Systems

[7] Y. Zhao et al., "DETRs beat YOLOs on real-time object detection," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2024.

[8] D. Das et al., "Outdoor trash detection in natural environment using a deep learning model," IEEE Access, vol. 11

[9] N. M. Roshni et al., "Trash detection in an aquatic environment," B.Sc. dissertation, BRAC University, Dhaka, Bangladesh, 2024.

[10] R. Harada et al., "Trash detection algorithm suitable for mobile robots using improved YOLO," Journal of Advanced Computational Intelligence and Intelligent Informatics

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learning Representations (ICLR), 2015.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)