Feature-Based Waste Classification System Technical Report

Classes: 6 primary (cardboard, glass, metal, paper, plastic, trash) + 1 unknown

---

1. Introduction

This report presents a feature-based computer vision system for automated waste classification. The system classifies waste materials into six categories using hand-crafted features and traditional machine learning classifiers (SVM and k-NN), achieving **84.1% accuracy with SVM** and **78.6% with k-NN**.

---

2. Data Augmentation

2.1 Augmentation Techniques

Seven augmentation techniques were applied to expand the training dataset.

Technique	Implementation	Justification
Horizontal Flip	`cv2.flip(image, 1)`	Items viewed from different angles on conveyor belt
Rotation ±15°	`cv2.warpAffine()`	Objects tilt and rotate during handling
Brightness +30%	HSV V-channel × 1.3	Variable lighting conditions (indoor/outdoor)
Brightness −30%	HSV V-channel × 0.7	Poor lighting, shadows from equipment
Zoom (80% crop)	Center crop + resize	Varying camera-to-object distances

Gaussian Noise (σ = 10)	Add N(0,10) noise	Camera sensor noise, dirty lenses

Why these specific techniques:

- **Horizontal Flip:** Preserves material properties while changing viewpoint.
- **±15° Rotation:** Prevents unrealistic distortions and deployment-inconsistent artifacts.
- **Brightness Adjustment (HSV):** Adjusting only the V-channel simulates lighting changes without color distortion.
- **Zoom:** Simulates depth variation due to camera distance.
- **Gaussian Noise (σ = 10):** Matches industrial camera noise characteristics.

Techniques not used:

- **Vertical Flip (unnatural orientation):** In real deployment scenarios (e.g., conveyor belts, sorting lines), waste items are not observed upside-down relative to gravity. Applying vertical flips would introduce physically implausible samples, forcing the model to learn orientations that will never appear at inference time. This can harm generalization by increasing intra-class variance without adding real-world robustness.
- **Extreme Rotations (>30°):** Large rotations often introduce black borders, interpolation artifacts, and unrealistic object poses that do not occur in controlled industrial environments where cameras are fixed. These artifacts can bias edge- and gradient-based features (especially HOG), reducing their discriminative power and leading to poorer decision boundaries.
- **Color Jitter (alters material appearance):** Color jitter modifies hue and saturation values, which are critical cues for material identification (e.g., brown cardboard, transparent glass, colored plastics). Altering these properties risks creating samples that violate material semantics (e.g., blue plastic appearing red), effectively introducing label noise and degrading classifier performance.

2.2 Augmentation Results

- **Target:** Minimum 1000 images per class
- **Achieved:** 965% increase (~10× for smallest class)

Class-wise augmentation:

- Cardboard: 197 → 956 images (4.9×)
- Glass: 306 → 957 images (3.1×)
- Metal: 253 → 967 images (3.8×)
- Paper: 357 → 943 images (2.6×)
- Plastic: 291 → 948 images (3.3×)
- Trash: 84 → 955 images (11.4×)

---

3. Feature Extraction

> **Design Iteration Note:** Multiple feature configurations were evaluated during development. Increasing feature dimensionality—particularly via higher-resolution HOG—initially appeared beneficial but resulted in longer SVM training times and degraded k-NN performance due to the curse of dimensionality and sometimes SVM performance too. The final configuration reflects a balanced, empirically validated design.

3.1 Feature Vector Architecture (193 Dimensions)

Feature Group	Dimensions	Percentage	Purpose
Color Histogram (HSV)	96	49.7%	Material color signature
Color Moments	9	4.7%	Color distribution statistics
LBP Texture	32	16.6%	Surface texture patterns
Edge Features	17	8.8%	Shape boundaries
Statistical Features	12	6.2%	Global appearance
HOG Descriptors	36	18.7%	Shape / structure (critical)
Hu Moments	7	3.6%	Shape invariants
Haralick (GLCM)	20	10.4%	Advanced texture
Gabor Filters	8	4.1%	Multi-scale texture
Frequency (FFT)	5	2.6%	Periodic patterns
Total	**193**	**100%**	

3.2 Feature Justifications

3.2.1 Color Histogram (HSV) — 96 Features

HSV separates color from brightness, improving robustness to illumination changes.

Material signatures:

- Cardboard: Brown hue, low saturation
- Glass: Low saturation, variable hue

- Metal: Low saturation, high value
- Paper: Low saturation, high value
- Plastic: High saturation, wide hue range

3.2.2 HOG (Histogram of Oriented Gradients) — 36 Features (Most Important)

- Image size: 64 × 64
- Cell size: 32 × 32 (2 × 2 cells)
- Orientations: 9
- Block normalization: L2-Hys

> **Design Rationale:** Higher-resolution HOG configurations (e.g., 128×128 with smaller cells) produced feature vectors exceeding 1700 dimensions. While these increased training cost substantially, they did not yield consistent validation gains and significantly harmed k-NN performance and sometimes also SVM. The selected configuration represents the optimal accuracy–efficiency trade-off.

Why HOG is critical:

- Distinguishes bottles, cans, and boxes via shape
- Robust to lighting changes
- Strong historical performance in object recognition

3.2.3 LBP (Local Binary Patterns) — 32 Features

Captures surface texture differences such as smooth plastic versus rough cardboard using rotation-invariant uniform patterns.

3.2.4 Haralick (GLCM) — 20 Features

Computed at distance = 1 over four angles to capture texture directionality and fine-grain patterns.

3.3 Feature Standardization

Z-score normalization was applied to all features.

- Prevents dominance of large-scale features
- Essential for SVM performance
- Without normalization, accuracy drops by 20–30%

---

4. Classifier Architectures

4.1 Support Vector Machine (SVM)

4.1.1 Theory and Architecture Design

The Support Vector Machine (SVM) classifier is designed to accept the **193-dimensional standardized feature vector** as input. Since SVM is inherently a binary classifier, a **one-vs-one (OvO)** strategy is employed to handle the six-class problem, resulting in **15 binary classifiers**. Each classifier learns a decision boundary between a pair of material classes, and final class prediction is obtained through majority voting.

This architecture is well-suited for high-dimensional, hand-crafted feature spaces, where SVMs can construct robust decision boundaries by maximizing the margin between classes.

4.1.2 Kernel Selection: RBF

The Radial Basis Function (RBF) kernel was selected due to the **non-linear separability** of waste material classes in the extracted feature space. Several classes (e.g., glass and plastic) exhibit overlapping color and texture characteristics, making linear separation insufficient.

The RBF kernel enables flexible non-linear decision boundaries by implicitly mapping feature vectors into a higher-dimensional space, while maintaining controlled model complexity.

4.1.3 Hyperparameter Selection and Justification

- **C = 5:** Controls the trade-off between margin maximization and classification error. A moderate value was chosen to limit overfitting while still enforcing effective class separation.
- **Gamma = 0.01:** Determines the influence radius of individual training samples. This value provides a balanced kernel smoothness for the 193-dimensional feature vector, avoiding overly complex or overly smooth decision surfaces.
- **Kernel = RBF:** Enables non-linear class discrimination.

These parameters were selected through empirical tuning and cross-validation, satisfying the requirement to justify optimal architectural elements.

4.1.4 Unknown Class Rejection Mechanism

Standard SVMs always assign a class label, even for samples that do not belong to any known category. To address this limitation, a **dual-threshold rejection mechanism** was implemented:

- **Confidence Threshold (0.4):** If the maximum predicted class probability is below this value, the classifier is considered uncertain.
- **Margin Threshold (0.5):** If the difference between the highest and second-highest class scores is small, the sample lies close to a decision boundary and is rejected.

A sample is labeled as *unknown* if either condition is met, preventing forced misclassification of out-of-distribution materials.

Rejection rate: 3.5%.

4.2 k-Nearest Neighbors (k-NN)

4.2.1 Architecture and Hyperparameter Selection

The k-NN classifier directly operates on the **193-dimensional standardized feature vectors** using a lazy learning approach, where classification is performed based on similarity to stored training samples.

Selected hyperparameters:

- **k = 7:** Provides a balance between sensitivity to local neighborhood structure and robustness to noise.
- **Weights = distance:** Assigns higher influence to closer neighbors, improving classification reliability.
- **Metric = Manhattan (L1):** Chosen for improved robustness in high-dimensional feature spaces, where Euclidean distance can become less discriminative.

These architectural choices were validated empirically through cross-validation.

4.2.2 Unknown Class Rejection Mechanism

Unlike SVM, k-NN does not provide explicit confidence estimates. Therefore, a **distance-based rejection strategy** was implemented:

- The median distance between training samples is used as a reference scale.
- A test sample is rejected if its average distance to the k nearest neighbors exceeds **2.5×** this median distance.

This relative distance criterion enables scale-invariant detection of samples that lie far outside the known feature distribution.

Rejection rate: 3.5%.

---

5. Architecture Comparison

5.1 Performance Summary

Metric	SVM	k-NN	Difference
Accuracy	84.1%	78.6%	+5.5%
Precision	84.1%	79.9%	+4.2%
Recall	84.1%	78.6%	+5.5%
F1-score	84.0%	78.7%	+5.3%
Inference Time	1.81 ms	0.66 ms	+1.15 ms
Rejection Rate	3.5%	3.5%	0.0%

5.2 Key Observations

- SVM consistently outperforms k-NN in accuracy
- Major confusion between glass and plastic
- Both systems meet real-time requirements
- Increasing feature dimensionality beyond the final design produced diminishing or negative returns
- k-NN performance degraded sharply in high-dimensional feature spaces, consistent with the curse of dimensionality

---